Learning-based Localizability Estimation for Robust LiDAR Localization

Author(s):
Nubert, Julian; Walther, Etienne; Khattak, Shehryar Masaud Khan; Hutter, Marco

Publication date:
2022

Permanent link:
https://doi.org/10.3929/ethz-b-000558164

Originally published in:
https://doi.org/10.1109/IROS47612.2022.9982257

Funding acknowledgement:
852044 - Learning Mobility for Real Legged Robots (EC)
101016970 - Natural Intelligence for Robotic Monitoring of Habitats (EC)
188596 - Perceptive Dynamic Locomotion on Rough Terrain (SNF)

Learning-based Localizability Estimation for Robust LiDAR Localization

Julian Nubert[†1,2], Etienne Walther[†1], Shehryar Khattak[1,2] and Marco Hutter[1]

***Abstract*— LiDAR-based localization and mapping is one of the core components in many modern robotic systems due to the direct integration of range and geometry, allowing for precise motion estimation and generation of high quality maps in real-time. Yet, as a consequence of insufficient environmental constraints present in the scene, this dependence on geometry can result in localization failure, happening in self-symmetric surroundings such as tunnels. This work addresses precisely this issue by proposing a neural network-based estimation approach for detecting (non-)localizability during robot operation. Special attention is given to the localizability of scan-to-scan registration, as it is a crucial component in many LiDAR odometry estimation pipelines. In contrast to previous, mostly traditional detection approaches, the proposed method enables early detection of failure by estimating the localizability on raw sensor measurements without evaluating the underlying registration optimization. Moreover, previous approaches remain limited in their ability to generalize across environments and sensor types, as heuristic-tuning of degeneracy detection thresholds is required. The proposed approach avoids this problem by learning from a collection of different environments, allowing the network to function over various scenarios. Furthermore, the network is trained exclusively on simulated data, avoiding arduous data collection in challenging and degenerate, often hard-to-access, environments. The presented method is tested during field experiments conducted across challenging environments and on two different sensor types without any modifications. The observed detection performance is on par with state-of-the-art methods *after* environment-specific threshold tuning[1].**

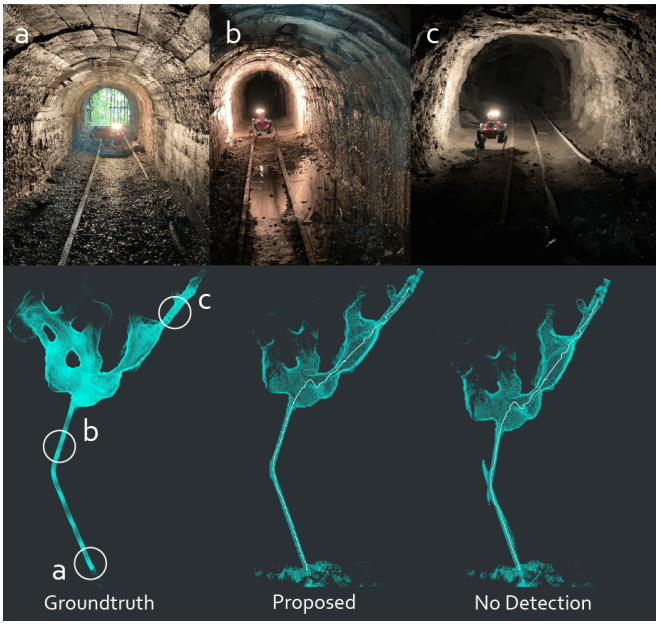

Fig. 1. Snapshots and maps from a challenging field test in an underground mine in Switzerland. The top-row images showcase different segments along the path that are a) localizable, b) non-localizable and c) mostly localizable. The bottom row shows the created map with and without the proposed localizability detection.

I. Introduction

Light Detection And Ranging (LiDAR) sensors are one of the most common sensors used for robot localization and mapping to date. The ability to provide depth measurements at long ranges and to operate under varying illumination conditions along with recent miniaturizing and reduction in costs have significantly increased their suitability for various robotic applications. To estimate robot pose from LiDAR data, popular techniques iteratively minimize the distance between point- or feature-correspondences in consecutive LiDAR scans to obtain the relative scan-to-scan pose transformations [1,2]. Furthermore, the individual LiDAR scans are aligned and merged to create a map representation of the environment, which facilitates further robot pose refinement and accurate map creation through scan-to-map [3] or scan-to-representation registration [4]. However, in order to converge to the correct solution these algorithms depend on sufficient geometric constraints in the environment. In the absence of such constraints, as in geometrically self-similar or symmetric environments, the optimization problem can become ill-conditioned and provide a sub-optimal solution. While the trend towards learned scan-to-scan estimation approaches [5,6] can redress the need of hand-crafting features, it can not eliminate the occurrence of such environment dependent "LiDAR slip", often occuring in corridors, hallways, tunnels or planar environments.

To mitigate such problems, most of the techniques focus on the exposure of ill-conditioned optimization problems. However, they remain limited in two aspects: first, most techniques make use of eigenvalues or the condition number of the approximate hessian matrix of the optimization problem. This, however, requires computationally expensive steps of data filtering and establishing point correspondences to be performed before the detection. Second, the detection of optimization degeneracy mostly relies on a heuristic threshold, which can vary significantly between different environments and sensors, hence limiting robot operation in degenerate environments without manual re-tuning of thresholds. Furthermore, it should be noted that most existing

This work is supported in part by the Max Planck ETH CLS, the EU Horizon 2020 programme grant agreement No.852044 and 101016970, the NCCR digital fabrication and robotics, and the SNSF project No.188596.

† Indicates equal contribution.

[1]The authors are with the Robotic Systems Lab, ETH Zürich.

[2]The authors are with the Max Planck ETH CLS.

Corresponding Author: Julian Nubert, nubertj@ethz.ch

[1]Supplementary Video: https://youtu.be/fm08PFwMO0c

methods that detect optimization degeneracy are dependent on both the target and reference point cloud scans and cannot evaluate the data quality provided by a single scan.

To overcome these challenges, this work proposes a method for learning-based localizability estimation from a single point cloud scan. The predicted metric encapsulates how localize-able a given point cloud would be w.r.t other point cloud scans, and can be used to predict upfront whether scan-to-scan registration will succeed for $3D$ pose estimation. Furthermore, the proposed approach is trained in an end-to-end manner using simulated sensor data only, and hence, avoids data collection in degenerate and often difficult-to-access environments. The experiments performed both in simulation and on real-world data demonstrate that the learned-approach can correctly predict localizability across a variety of environments without a need of re-tuning, is capable to work with real data, and can even generalize to different LiDAR types. The main contributions of this work are as follows: i) The development of a learning-based approach that successfully predicts the LiDAR localizability in 6-DOF directly from point cloud data. ii) A thorough evaluation conducted to improve the robot localization performance of the ANYmal-C quadruped robot operating in challenging and degenerate real-world environments. iii) A thorough design and implementation of all components, including the used dataset and dataset generation. All relevant parts will be made publicly available as an open-source package for the benefit of the robotics community[2].

II. Related Work

Pose uncertainty estimation is an essential component of the robot state-estimation process, as it not only provides a quantitative measure of the quality of the pose estimates provided by a sensing modality, but also facilitates reliable sensor fusion. For LiDAR-based pose estimation, the works in [7,8] propose a closed-form estimate for Iterative Closest Point (ICP) methods by carefully analyzing the effect of incorrect convergence, under-constrained situations, and sensor noise on the error function. However, as discussed in [9], this closed-form formulation tends to be over-optimistic with regard to that pose uncertainty, and building on this insight, the authors in [10] propose to not ignore the effect of initialization accuracy on the final pose uncertainty. To avoid overconfidence and to include all potential sources of errors, multiple methods [11,12] employ sampling approaches to achieve a better ICP pose uncertainty estimate at the expense of additional computation. These formulations try to include all potential sources of errors under a single quality metric, which can be beneficial for sensor fusion tasks. However, such approaches do not provide an independent understanding of the quality of geometric constraints provided by the environment, which may be the primary cause of optimization ill-conditioning in challenging environments.

To detect optimization degeneracy in cases where environment do not provide enough geometric constraints, [13] proposes a method relying on eigenvalues to detect ill-conditioning and proposes to perform solution remapping along the degenerate directions. This approach has proven to perform well in various real world scenarios, e.g. in [14], however, the performance of the approach relies on heuristically defined eigenvalue thresholds that are highly dependent on the environment and the deployed sensor suite. Similarly, [15] relies on the condition number to determine the health of the optimization process and includes partial constraints along non-degenerate direction for sensor fusion. Other methods, such as [16,17] rely on the final alignment of scans to capture the adequacy of geometric constraints provided by the environment for correct solution convergence. However, all these methods either rely on the point cloud registration process or its result to determine the performance of the pose estimation process, and do *not* exploit the information provided by the point cloud data directly to facilitate the estimation process itself. Addressing this issue, [18,19] propose to quantify the geometric constraints provided by point cloud scan or map data and suggest to focus on a localizability metric instead of degeneracy detection. In [18], the authors demonstrate measuring localizability of a point cloud map by evaluating constraints provided by the data, while in [19] they use a similar approach to perform sensor fusion when insufficient constraints are present during the traversal along a tunnel-like environment. In a similar manner, [20] proposes to segment the map first into geometric features and then evaluates whether the combined constraints provided by all features are sufficient for determining the 6-DOF pose accurately. However, one of the main limitations of all these methods is the determination of a heuristic threshold that is highly dependent on the environment and difficult to pre-determine in a reliable manner.

Arguing along similar lines, [21] proposes to employ machine learning techniques to predict robot pose uncertainty by directly learning from sensor data. The proposed technique demonstrates learning-based $2D$ pose covariance prediction by utilizing planar LiDAR features. Extending the same concept to $3D$, authors in [22] demonstrate sensor data-driven learning methods for scan-to-scan matching of $3D$ LiDARs. Although these methods demonstrate the heuristic-free advantage of learning-based methods, they still rely on careful extraction and selection of features from sensor data, rendering them sensitive to the feature selection process and limit the utilization of all available sensor data. To utilize the raw sensor outputs, [23] demonstrates the application of an end-to-end deep-learning method to predict the pose estimation uncertainty of an autonomous underwater vehicle directly from $3D$ bathymetric point clouds. However, the given method is limited to $2D$ robot navigation in specific underwater scenarios, leaving a gap to higher dimensional and more complex robotic systems.

Given the discussion above, this work fills a gap by proposing a heuristic-free learning-based 6-DOF localizability estimation approach. The proposed approach is capable of operating directly on LiDAR scan data in an end-to-end manner and avoids the need for developing hand-crafted

[2] https://github.com/leggedrobotics/L3E

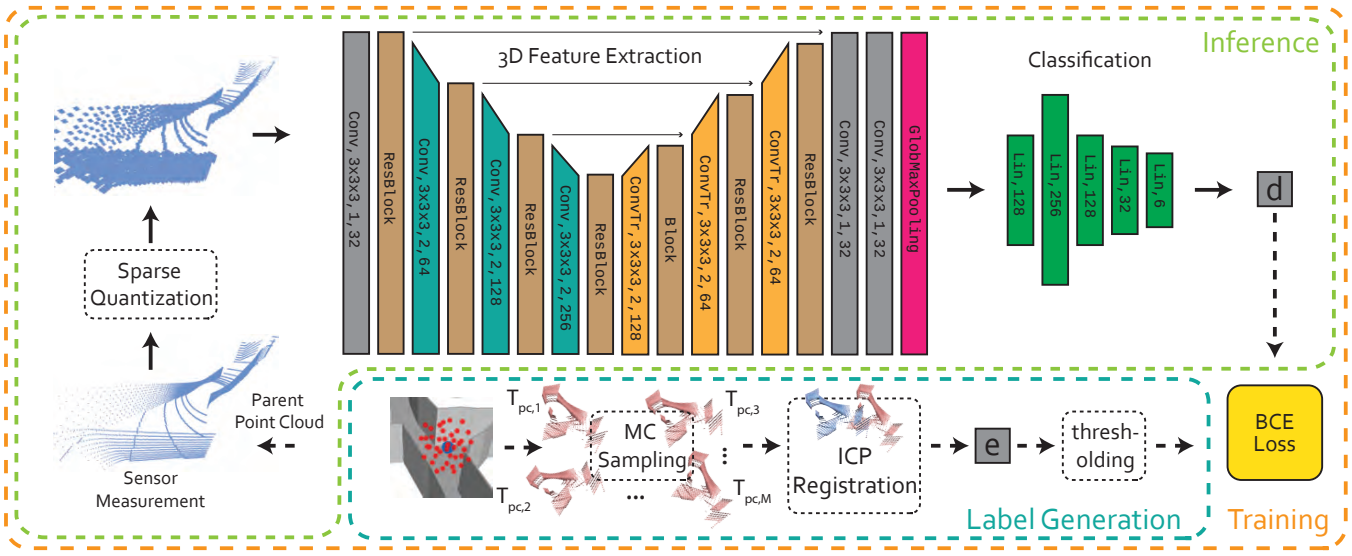

Fig. 2. Overview of the proposed approach. The input point cloud received from the LiDAR sensor is down-sampled, sparsely quantized, and then fed into a sparse $3D$ convolutional neural network based on the ResUnet architecture. Monte Carlo sampling and ICP point cloud registration are used to generate the ground truth labels required for the supervision signal in order to train the feature extraction and classification network in an end-to-end fashion.

features. The proposed technique is fully trained in simulation, and avoids the difficult task of data collection in challenging geometrically degenerate environments, and is capable to generalize to a variety of real-world environments without tuning or modification.

III. Problem Formulation

Given the aforementioned discussion, the goal of this work is to reliably predict the ability to successfully localize in certain directions using scan-to-scan registration in the current environment. For this purpose, this work aims to provide a localizability estimate directly from a $3D$ LiDAR point cloud scan $\mathbf{s}_k \in \mathbb{R}^{n_k \times 3}$. Conceptually, localizability can be thought of as a detection measure along 6-DOF, which in this work is defined as

$$\mathbf{d}_k = (d_x, d_y, d_z, d_\phi, d_\theta, d_\psi)^\top, \tag{1}$$

with x, y, z denoting translation coordinates, and ϕ, θ and ψ denoting the Euler angle representation for roll, pitch and yaw, respectively. For practical reasons, the localizability is considered binary; either a certain direction is localizable or not. Hence, it holds $d_i \in \{0, 1\}\ \forall\ i \in \{x, y, z, \phi, \theta, \psi\}$, where 0 (localizable) and 1 (non-localizable) express whether or not the given direction can be localized successfully in the current environmental surroundings via scan-to-scan LiDAR registration of $\mathbf{s}_k$ against scan $\mathbf{s}_{k-1}$.

The estimation of $\mathbf{d}_k$ is formulated as a multi-label binary classification problem, and is performed through a neural network classifier. The localizability at time k is modelled to be fully determined by the current point cloud scan observation $\mathbf{s}_k \in \mathbb{R}^{n_k \times 3}$. The resulting function $\mathbf{s}_k \rightarrow \mathbf{d}_k$ can hence be approximated through $\tilde{\mathbf{d}}_k(\Theta, \mathbf{s}_k)$, where $\Theta \in \mathbb{R}^P$ are the P trainable parameters of the network. These parameters are obtained via minimization of a supervised classification loss $\underset{\Theta}{\operatorname{argmin}}\, \mathcal{L}(\tilde{\mathbf{d}}(\Theta, \mathcal{S}), \mathcal{T})$ over a training set of scans $\mathbf{s}_i \in \mathcal{S}$ with ground truth labels $\mathbf{t}_i \in \mathcal{T}$.

IV. Proposed Approach

As introduced in Section III, the goal of this work is to reliably determine the localizability measure $\mathbf{d}_k$ at time k during robot operation by only considering the current LiDAR scan $\mathbf{s}_k$. This section details the proposed approach. An overview of the steps taken is shown in Figure 2.

A. Localizability Measure

The specific definition of the localizability measure $\mathbf{d}_k$, e.g. needed for the training-data generation, requires two steps: first an expected registration error $\mathbf{e}_k \in \mathbb{R}^6$ is computed, similarly to [23]. Second, this registration error is mapped to the desired metric $\mathbf{d}_k$ through a thresholding operation.

1) Expected Registration Error: In order to compute the expected registration error $\mathbf{e}_\mathrm{p}$ for a *parent point cloud* $\mathbf{s}_\mathrm{p}$, the average registration residual of a defined point cloud distribution around $\mathbf{s}_\mathrm{p}$ needs to be determined.

Given $\mathbf{s}_\mathrm{p}$, the first step is to perform Monte Carlo sampling for a collection of M *child point cloud* scans $\mathbf{s}_{\mathrm{c},j}, j \in \{1, ..., M\}$ in proximity of the parent cloud. To do so, the required child point cloud poses $\mathbf{T}_{\mathrm{p,c},j} \in SE(3)$ are computed by drawing 6 perturbations in directions $i \in \{x, y, z, \phi, \theta, \psi\}$ from zero-mean Gaussians $\mathcal{N}(0, \sigma_i^2)$ with parameters given in Table I, and converting them to $SE(3)$. The selected sampling parameters are chosen according to the maximum expected motion of the robot between two consecutive scans. Each resulting child point cloud is then registered against the parent point cloud using point-to-plane ICP. The expected registration error $\mathbf{e}_\mathrm{p}$ for parent point cloud $\mathbf{s}_\mathrm{p}$ is computed as the mean absolute error:

$$\mathbf{e}_\mathrm{p} = \frac{1}{M} \sum_{j=1}^{M} \left| \gamma \left(\tilde{\mathbf{T}}_{\mathrm{pc},j}^{-1} \cdot \mathbf{T}_{\mathrm{pc},j} \right) \right|. \tag{2}$$

Here, $\tilde{\mathbf{T}}_{\mathrm{pc},j} \in SE(3)$ denotes the estimated transformation from the registration algorithm, and γ denotes the projection

back from $SE(3)$ into $(x, y, z, \phi, \theta, \psi)$. Note, that due to the

TABLE I
PARAMETERS CHOSEN FOR PERFORMING THE MONTE CARLO SAMPLING.

M	σ_x[m]	σ_y[m]	σ_z[m]	σ_ϕ[°]	σ_θ[°]	σ_ψ[°]
200	0.1	0.1	0.1	5.0	5.0	10.0

sparse and circular structure of LiDAR scans, in contrast to the approach in [23], scans $\mathbf{s}_\mathrm{p}$ and $\mathbf{s}_{\mathrm{c},j}$ can not be the same perturbed scan. Instead, an additional point cloud scan must be sampled at every child pose.

2) Obtaining Localizability Measure: After computing the $6D$ registration errors, the binary labels in $\mathbf{d}_k$ can be obtained by applying thresholds to the expected registration error:

$$\mathbf{d}_k = \alpha(\mathbf{e}_k). \tag{3}$$

Here, α is a component-wise thresholding operation, with the motion thresholds as presented in Table II. These values are

TABLE II
MOTION THRESHOLDS FOR COMPUTING $\mathbf{d}_k$ FROM $\mathbf{e}_k$.

e_x[m]	e_y[m]	e_z[m]	e_ϕ[°]	e_θ[°]	e_ψ[°]
0.1	0.1	0.1	2.0	2.0	2.0

selected according to the sampling parameters from Table I, considering a less than 10cm error in translation and 2° in rotation as indication of convergence.

B. Architecture

The choice of the right architecture and data representation is one of the key building blocks when developing a neural network estimation method.

1) Data Representation & Feature Extraction: Compared to neural network architectures deployed for image processing, with $2D$ convolutional neural networks being the prominent choice, the selection of the best network architecture for processing $3D$ point cloud data is often an open question. Commonly chosen architectures include but are not limited to: set-based methods [24], graph-based approaches [25], image-view techniques, and voxelization-based architectures [26]. While projection-based image-view representations have been chosen frequently for rotating LiDAR sensors [5,6,27] due to low memory requirements and the possibility to use well-explored $2D$ convolutional neural networks, generalization to different sensor types with varying field of view, intensity characteristics, or number of vertical rays remains difficult. In a related work [23], the authors propose to make use of the PointNet architecture to predict $2D$ localization covariances. While these set-based methods have shown to perform well for global classification tasks, they were specifically designed to be invariant to rigid body transformations. Often deemed to be memory hungry and inefficient, in recent years voxel-based $3D$ convolutional neural networks gained importance through the rise of sparse convolutions [28]. Motivated by these developments, this work builds on an architecture based on sparse $3D$ convolutions, allowing the network to develop a good scene understanding while remaining fast and memory efficient. To justify this network choice, a comparison against the PointNet architecture for the localizability prediction task is presented in Section VI-A.

A full overview of the utilized network architecture is presented in Figure 2. The network design is based on the $3D$ ResUNet [29] architecture and is inspired and adopted from [30], within which state-of-the-art results for point cloud feature learning are demonstrated. Given a LiDAR point cloud provided as an input, first a sparse quantization is performed to down-sample the point cloud to the correct density and to obtain the correct data-format. The architecture is generally built up like a UNet but with a ResNet block consisting of two convolutional networks plus a skip connection after each down- and up-sampling convolutional layer. Global max pooling is applied at the end in order to get a constant-size feature vector of size 32.

2) Classification Network: The output of the previous feature extracting network is then forwarded to an MLP with 5 layers. The final output of the network is a $6D$ vector. After channeling each of the entries through a sigmoid activation function, a probability estimate $\tilde{\mathbf{p}}_k$ is obtained.

C. Loss

Formulating the objective as a classification problem circumvents addressing, in part, the very different scales of translation and rotation components. Binary cross entropy is chosen to be the loss function during training, which for the raw probability vector $\tilde{\mathbf{p}}_k$ is defined as

$$\mathcal{L}_k = -\frac{1}{6}\sum_{i=1}^{6} t_{k,i} \cdot \log(p_{k,i}) + (1 - t_{k,i}) \cdot \log(1 - p_{k,i}). \tag{4}$$

Here, $t_{k,i}$ denotes the target label obtained during the procedure described before.

D. Classification Threshold

Instead of dealing with hard-to-interpret eigenvalue metrics, which often require manual tuning of detection thresholds, the output of the proposed network has the characteristics of a probability estimate for each of the classes. To transform the

TABLE III
PROBABILITY THRESHOLDS FOR DETERMINING $d_{k,i}$ FROM $p_{k,i}$.

p_x[m]	p_y[m]	p_z[m]	p_ϕ[°]	p_θ[°]	p_ψ[°]
0.3	0.3	0.8	0.3	0.8	0.8

provided probabilities into class labels, a natural but naïve choice is to threshold each of the entries of $\tilde{\mathbf{p}}_{k,i}$ at 50%. Instead, in this work the thresholds are chosen according to a *precision-recall* curve generated for the validation set. The same thresholds from Table III are used for all experiments within this work.

E. Training Procedure

For efficient training of the proposed network, some additional design consideration are taken into account.

1) Dataset Balancing: Although care has been taken during the dataset generation, the dataset is still highly imbalanced with regard to positive and negative instances for each of the six directions. In particular, there are very few instances in the dataset that are non-localizable for z, ϕ and θ. To reduce this imbalance, so called *power-set labels* [31] are used to balance the dataset. Overall, $2^6 = 64$ such labels are generated, where up-sampling is performed with respect to the second most common class. The maximum up-sampling factor is set to be 100 in order to limit the over-sampling of heavily underrepresented power-set-labels.

2) Training Characteristics: For training the network, stochastic gradient descent (SGD) with a batch-size of 8 is selected. The learning rate is chosen to decay at an exponential rate of $\gamma = 0.9$. To avoid over-fitting, dropout is utilized in the classification network. Further, 4000 random points in the point cloud scan are sampled during training and deployment, followed by sparse quantization with a voxel-size of 0.2m.

V. Data Generation & Implementation

This section details the environment generation and practical data sampling procedure according to the mathematical foundations laid out in Section IV-A. Furthermore, implementation details of the network are provided.

A. Dataset Generation

As introduced in Section IV, the ground truth labels are generated via Monte Carlo sampling of child point clouds in a neighborhood of parent point clouds. In order to obtain meaningful scan-to-scan registration results that can be used for computing the expected registration error $\mathbf{e}_k$, more than just rotating and translating the parent point cloud is required to obtain suitable child point clouds. Instead, both source and target, i.e. parent and child point cloud, have to be recorded through ray-casting at the sampled sensor location (in simulation or reality). To render this possible, and to coincide with the theoretical sampling procedure introduced earlier in Section IV-A, this work performs the whole scan collection purely in simulation with the LiDAR sensor being spawned at parent and child poses.

1) Environments: All point clouds are sampled in simulation from a collection of environments. The environment selection and generation is done in a way to represent as many of the basic types of degenerate and non-degenerate cases as possible. As a result, the created dataset contains instances of planes, tunnels, cylinders, and meshed cave environments. A small collection of these environments is shown in Figure 3 on the left. Overall, 15 environments were collected for the final training and validation dataset generation. The used meshes are either obtained purely from CAD or by meshing down-sampled point cloud scans of real cave environments. All environments are made publicly available[2].

2) Point Cloud Sampling in Simulation: To sample a diverse set of parent point clouds, three measures are taken during the sampling process. First, in order to efficiently sample meaningful point clouds from inside the environments, *sampling paths* consisting of poly-lines are hand-drawn along the ground of the meshes using Cloud Compare[3]. An example of such a sampling path is shown in yellow in Figure 3 on the right. Second, the parent point cloud sampling is performed in proximity to the sampling paths in Gazebo, with the point clouds being recorded using the simulated model of a Velodyne VLP-16 LiDAR. Each sensor pose in the world frame $T_{\texttt{wp}}$ is sampled around selected points of the sampling path according to normal distributions with zero mean and the following variances: $\sigma_{\text{trans},x,y} = 0.2$m, $\sigma_{\text{trans},z} = 0.4$m, $\sigma_{\text{rot},\phi,\theta} = 15°$, $\sigma_{\text{rot},\psi} = 180°$. Third, collision checking for every resulting transformation $T_{\texttt{wp}}$ is performed, and if collision-free, the child point locations are sampled around the parent pose as described in Section IV-A.

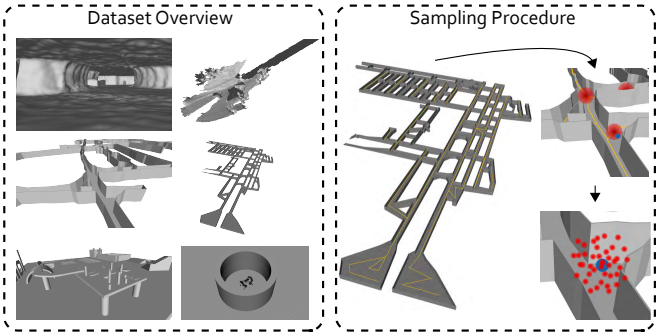

Fig. 3. Left: an overview of some of the used environments is shown, mainly created in CAD or sampled from underground environments. Right: an exemplary sampling path is depicted for collecting parent point clouds.

B. Network Implementation

The $3D$ convolutional neural network is implemented using PyTorch and MinkowskiEngine [28]. The deployed network can be trained on a Nvidia RTX3090 GPU in about 4 hours. The model which is deployed during the experiments in Section VI was trained for 50 epochs. Without any specific code optimization, the inference time of the PyTorch model embedded into a ROS node is 28 milliseconds for CPU-only mode (Intel i7 10700F) and 13 milliseconds when being deployed on a Nvidia RTX3070 GPU.

VI. Experimental Results

To evaluate the feasibility of the proposed approach and to demonstrate its suitability towards real-world applications, a set of experimental studies are conducted. First, the choice of the network architecture is validated through an ablation study. Next, the potential of the proposed approach is demonstrated through robotic field experiments conducted in three different environments, with the first two being highly degenerate. Last, the generalization of the presented method is demonstrated by showing that the same trained network can work across two different LiDAR sensors, varying in their field-of-view and point cloud density, leading to similar performance when being deployed in the same environment.

[3] https://www.cloudcompare.org/main.html

A. Network Architecture Validation

To verify the suitability of using ResUNet over PointNet (as is used in [23]), the architectures' performance is studied using the same training, validation and test sets. To that end, the performance on the training set is assessed for a better understanding of the expressiveness of the network, while the validation set is studied to assess the actual learning capability in familiar but unseen scenarios. Furthermore, a test set is created in simulation from a ground truth mesh of the tunnel environment (cf. Figure 1, 4), which is used to numerically compare the generalization abilities of the two models. The corresponding ground truth map was recorded using *Leica RTC360* and *BLK2GO* sensors. Table IV provides insights into the performance of the two architectures trained for 60 epochs, across the three described scenarios by listing accuracy, F1-score, precision, and recall. The shown values are the averaged values of the six dimensions $\{x, y, z, \phi, \theta, \psi\}$.

TABLE IV

COMPARISON OF RESUNET AND POINTNET FEATURE EXTRACTORS. THE SHOWN VALUES ARE AVERAGED VALUES OVER ALL SIX DIMENSIONSS.

	ResUNet (proposed)			PointNet		
	Train	Valid	Test	Train	Valid	Test
Accuracy	**0.998**	**0.961**	**0.854**	0.957	0.945	0.809
F1-score	**0.997**	**0.606**	**0.517**	0.94	0.53	0.214
Precision	**0.997**	**0.647**	**0.398**	0.918	0.472	0.22
Recall	**0.998**	0.585	**0.752**	0.959	**0.703**	0.244

It can be noted that the ResUnet architecture outperforms PointNet in all evaluation scenarios except the recall on the validation set. The much better performance on the training set indicates more capacity to represent certain distributions. While this can potentially result in over-fitting, ResUNet also outperforms PointNet on the validation set. Finally, the performance of the proposed architecture is superior on the test set; when considering F1 score, precision and recall, the practical advantage of ResUNet over PointNet is underlined. Note, that the numeric result of precision and recall are higher for x, y, ψ, while being relatively low for z, ϕ, θ as a consequence of few positive examples in the training set.

B. Field Experiments

To evaluate the real-world performance of the proposed method, a set of field deployments were conducted using the ANYmal quadrupedal robot [32] equipped with a Velodyne VLP-16 LiDAR. The field tests were conducted in: i) a tunnel environment located in an underground mine, ii) an open field at a paved work site, and iii) an indoor urban environment containing narrow corridors and open spaces, offering a variety of challenging scenarios for LiDAR-based localization due to self-symmetry and lack of geometric constraints along different directions.

To demonstrate the contribution of the proposed approach towards improving robot localization for operation in challenging environments, the localizability predictions of the proposed approach are integrated as a degeneracy source into a complementary multi-modal localization and mapping framework, CompSLAM [33]. This framework was used by team CERBERUS [14] during their winning run of the DARPA Subterranean challenge, and is capable of handling single estimation failures through multi-modal sensor fusion. For the purpose of this work, the proposed method replaces the LiDAR degeneracy detection module in CompSLAM, which is originally based on the work of [13]. Hence, in-addition to evaluating the impact on real-world robot localization performance, this also allows for a comparison against current state-of-the-art LiDAR degeneracy detection methods.

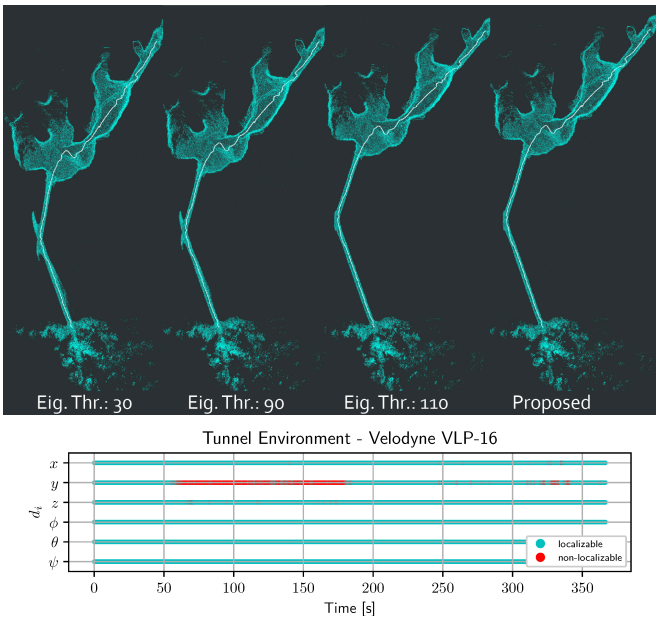

Fig. 4. Predicted localizability and mapping results for the tunnel environment. Maps for three different eigenvalue thresholds $\{30, 90, 110\}$ as well as the proposed approach are presented. While an e-value threshold of at least 110 is required to reliably detect degeneracy, the network detects the degeneracy along the robot's y-axis without any intervention.

1) Tunnel Environment: To evaluate the performance of the proposed method in a real-world LiDAR degenerate environment, a field test was conducted at Seemühle mine, Switzerland. This abandoned mine has a long tunnel corridor connecting the entrance to the main quarry area as shown in the ground truth scan in Figure 1. During this experiment, the ANYmal-C robot started outside the mine and traversed towards the quarry. Both these areas have enough geometrical features for LiDAR-based localization to function properly, however, the connecting tunnel has smooth walls and is symmetric along its principal axis, representing a practical failure scenario. Accurate prediction of localizability is essential for robot operation in this environment to provide reliable robot localization through sensor-fusion. In this experiment, kinematic leg-odometry based on work of [34] is fused with LiDAR localization in CompSLAM along the predicted non-localizable DOF, in order to avoid localization failure and to produce an accurate map of the environment, as shown in Figure 1, 4. It should be noted that the proposed method is not trained in this environment, but only on sampled generated in simulation. Furthermore, to demonstrate the advantage of the heuristic-tuning-free nature of the proposed method, the

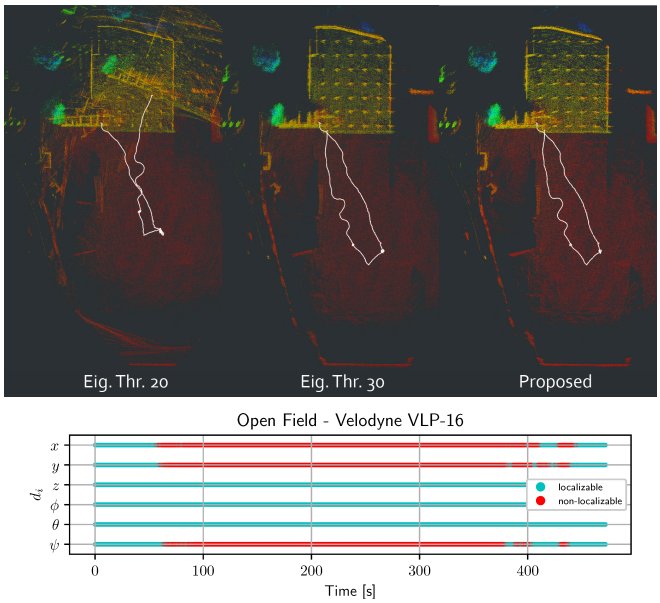

Fig. 5. Maps for e-value thresholds of $20, 30$, and localizability results for the open field experiment. In this scenario, an eigenvalue threshold of 30 is sufficient to detect the degeneracy.

deterioration effects of using incorrect heuristic thresholds on robot localization and mapping in such challenging environments is shown in the top of Figure 4. Different eigenvalue thresholds are set for the default degeneracy detection approach [13] used in CompSLAM and it can be observed that the map quality is sensitive to even small changes in the thresholds used, corroborating the need for heuristic-free approaches. In contrast, the lower of Figure 4 shows the proposed approach detecting the degeneracy along the robot's y-axis correctly, which allows for a non-corrupted map creation without parameter tuning.

2) Open Field: Utilizing the same experimental setup with the same trained network from the previous section, a field robotic deployment was conducted on an open field near a work site in Rümlang, Switzerland. During this experiment, the robot starts underneath a canopy, and then traverses over a large concrete field without any geometric features nearby, causing the LiDAR-based scan-to-scan registration to become degenerate along x, y and ψ (yaw) directions, as also predicted by the network in lower Figure 5. The proposed method can predict the localizability correctly resulting in an accurate map of the environment to be built, as shown in the top of Figure 5. Furthermore, it can be noted that a different eigenvalue threshold needs to be set for the degeneracy detection method to work properly as compared to the experiment in the tunnel environment, whereas the same trained network is deployed across both experiments demonstrating the potential of the proposed method to work across different real-world and challenging environments.

3) Urban Environment: Towards evaluating the performance of the proposed method in urban environments, a robotic experiment was conducted in an office-like environment at ETH Zürich, Switzerland. This environment contains narrow corridors, glass surfaces and open spaces both in

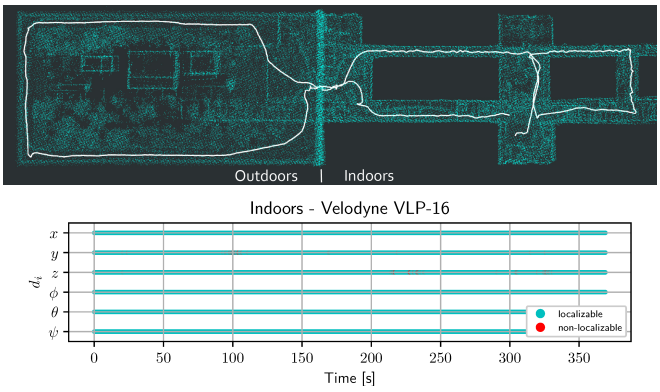

Fig. 6. Created map and localizability results in the office environment. Localizability is detected to be present for this scenario, so the full scan-to-scan registration is used as a prior to the mapping.

indoor and outdoor settings. During the experiments, the robot starts indoors and traverses outdoors on a roof-top terrace before looping back to the start position. The robot path and the generated environments map can be seen in upper Figure 6. In the lower plot it can be noted that the proposed approach provides the correct localizability estimates during this experiment, mostly localizable, validating it as a potential solution for operation across a variety of environments.

C. Generalization

To evaluate the generalization capabilities of the proposed approach, experiments are performed with a different LiDAR sensor type, whose measurements have not been observed during training. For this experiment, a similar robot path is traversed through the same underground mine tunnel presented in Section VI-B.1, with the robot being equipped with a different LiDAR sensor; a 128-beam Ouster OS0 LiDAR with a 90 degree vertical field-of-view. This represents a big step from the 16-beam Velodyne VLP-16 sensor with only a 30 degree vertical field-of-view that was used during training and previous experiments. This sensor variation imposes a large change on the environment observation during

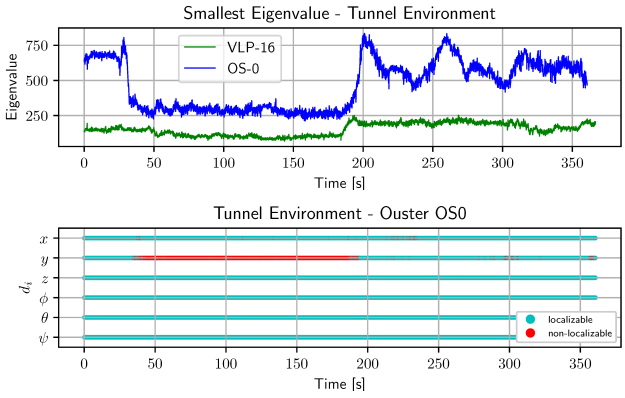

Fig. 7. Comparison of the smallest eigenvalues of the detection approach in [13] for Velodyne VLP-16 and the Ouster OS0-128 LiDAR sensors. Although traversing the same environment, the scale of the eigenvalues differs significantly. Despite this, the proposed network detects non-localizability reliably without any refinement.

a single scan and a significant increase in the input point cloud data. To illustrate the difference quantitatively, the smallest eigenvalues, i.e. the ones traditionally used for degeneracy detection, are shown for both sensor types along the path in upper Figure 7: The scale of the optimization problem is impacted significantly, necessitating heuristic threshold to be re-tuned for this experiment. However, comparing the bottom plots in Figure 4 and Figure 7, one can conclude that the proposed approach's estimation ability is not significantly impacted by the sensor change, and is still able to correctly predict localizability for both sensor types.

VII. Conclusions & Future Work

This work presented an end-to-end trained approach for localizability estimation of LiDAR-based point cloud registration. The presented approach is purely trained on simulated data, allowing for an easy scale-up of the generated training data, if needed. Suitability of the selected architecture and training procedure is shown through an ablation study. Practicality and the ability to generalize to real world robotic use-cases is demonstrated through three experiments conducted in i) a self-symmetric tunnel and cave mission, ii) a large scale planar outdoor scenery, and iii) in an indoor office environment. The same trained network was deployed across all experiments without the requirement of heuristic threshold tuning. For future work, one promising direction of research is to extend the presented approach to predict a full 6-DOF covariance matrix $\mathcal{M} \in \mathbb{R}^{6\times6}$, which would allow for precise localizability estimation in scenarios where the robot is misaligned with the environment. Further plans include making use of this degeneracy information in the scan-to-map registration process as well. Finally, localizability detection of different sensor modalities is expected to help towards more robust and reliable sensor fusion, e.g. in the form of partial factors in optimization-based approaches [15,35].

References

[1] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp." in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.

[2] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3d-ndt," *Journal of Field Robotics*, vol. 24, no. 10, pp. 803–827, 2007.

[3] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Autonomous Robots*, vol. 41, no. 2, pp. 401–416, 2017.

[4] J. Behley and C. Stachniss, "Efficient surfel-based slam using 3d laser range data in urban environments," in *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[5] J. Nubert, S. Khattak, and M. Hutter, "Self-supervised learning of lidar odometry for robotic applications," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[6] Q. Li *et al.*, "Lo-net: Deep real-time lidar odometry," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8473–8482.

[7] A. Censi, "An accurate closed-form estimate of ICP's covariance," pp. 3167–3172, 2007, ISBN: 1424406021.

[8] ——, "On achievable accuracy for range-finder localization," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 4170–4175.

[9] S. Bonnabel, M. Barczyk, and F. Goulette, "On the covariance of icp-based scan-matching techniques," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 5498–5503.

[10] M. Brossard, S. Bonnabel, and A. Barrau, "A new approach to 3d icp covariance estimation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 744–751, 2020.

[11] T. Iversen, A. Buch, and D. Kraft, "Prediction of ICP pose uncertainties using monte carlo simulation with synthetic depth images," in *IEEE International Conference on Intelligent Robots and Systems*, 2017.

[12] F. A. Maken, F. Ramos, and L. Ott, "Estimating motion uncertainty with bayesian icp," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8602–8608.

[13] J. Zhang, M. Kaess, and S. Singh, "On degeneracy of optimization-based state estimation problems," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 809–816.

[14] M. Tranzatto *et al.*, "Cerberus: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the darpa subterranean challenge," *arXiv preprint arXiv:2201.07067*, 2022.

[15] A. Hinduja, B.-J. Ho, and M. Kaess, "Degeneracy-aware factors with applications to underwater slam," in *International Conference on Intelligent Robots and Systems*. IEEE, 2019.

[16] H. Almqvist, M. Magnusson, T. P. Kucner, and A. J. Lilienthal, "Learning to detect misaligned point clouds," *Journal of Field Robotics*, vol. 35, no. 5, pp. 662–677, 2018.

[17] D. Adolfsson, M. Magnusson, Q. Liao, A. J. Lilienthal, and H. Andreasson, "Coral–are the point clouds correctly aligned?" in *2021 European Conference on Mobile Robots (ECMR)*. IEEE, 2021, pp. 1–7.

[18] W. Zhen, S. Zeng, and S. Soberer, "Robust localization and localizability estimation with a rotating laser scanner," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 6240–6245.

[19] W. Zhen and S. Scherer, "Estimating the localizability in tunnel-like environments using lidar and uwb," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.

[20] Y. Liu, J. Wang, and Y. Huang, "A localizability estimation method for mobile robots based on 3d point cloud feature," in *IEEE International Conference on Real-time Computing and Robotics*. IEEE, 2021.

[21] 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.

[22] D. Landry, F. Pomerleau, and P. Giguere, "Cello-3d: Estimating the covariance of icp in the real world," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.

[23] I. Torroba, C. I. Sprague, N. Bore, and J. Folkesson, "Pointnetkl: Deep inference for gicp covariance estimation in bathymetric slam," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4078–4085, 2020.

[24] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[25] W. Shi and R. Rajkumar, "Point-gnn: Graph neural network for 3d object detection in a point cloud," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[26] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *International Conference on Intelligent Robots and Systems*. IEEE, 2015.

[27] L. Stanislas, J. Nubert, D. Dugas, J. Nitsch, N. Sünderhauf, R. Siegwart, C. Cadena, and T. Peynot, "Airborne particle classification in lidar point clouds using deep learning," in *Field and Service Robotics*. Springer, 2021, pp. 395–410.

[28] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[29] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018.

[30] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8958–8966.

[31] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multilabel classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1079–1089, 2011.

[32] P. Fankhauser and M. Hutter, "Anymal: a unique quadruped robot conquering harsh environments," *Research Features*, 2018.

[33] S. Khattak *et al.*, "Complementary multi–modal sensor fusion for resilient robot pose estimation in subterranean environments," in *IEEE International Conference on Unmanned Aircraft Systems*, 2020.

[34] M. Bloesch, M. Burri, H. Sommer, R. Siegwart, and M. Hutter, "The two-state implicit filter recursive estimation for mobile robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 573–580, 2017.

[35] J. Nubert, S. Khattak, and M. Hutter, "Graph-based multi-sensor fusion for consistent localization of autonomous construction robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.