Generalizable Stable Points Segmentation for 3D LiDAR Scan-to-Map Long-Term Localization

Ibrahim Hroob[1,⋆], Benedikt Mersch[2,⋆], Cyrill Stachniss[2], and Marc Hanheide[1]

***Abstract*—Mobile robots increasingly operate in real-world environments that are subject to change over time. Accurate and robust localization is, however, crucial for the effective operation of autonomous mobile systems. In this paper, we tackle the challenge of developing a generalizable learned filter for long-term localization based on scan-to-map matching, using only 3D LiDAR data. Our primary objective is to enhance the reliability of mobile robot localization in dynamic environments. To obtain a strong generalization capability of the learned filter, we exploit the discrepancy between scan and map data. Our approach involves applying sparse 4D convolutions on a joint sparse voxel grid that encompasses both, scan voxels and their corresponding map voxels. This allows us to segment scan points into stable and unstable points based on a predicted long-term stability confidence score for each scan point. Our experimental results demonstrate that utilizing the stable points for localization improves the performance of scan-matching algorithms, especially in environments where changes in appearance are frequent. By exploiting the discrepancy between scan and map voxels, we enhance the segmentation of stable points. As a result, our approach generalizes to new, unseen environments.**

***Index Terms*—Object Detection, Segmentation and Categorization; Localization; Field Robots**

I. Introduction

EFFECTIVE localization is key for robot operation in many domains. Often, robots use a map as reference data and require their sensor readings against this map to localize themselves. Outdoor environments, however, may undergo significant changes over time, which imposes challenges for onboard vehicle localization systems. Achieving robust and accurate localization in such environments is a fundamental capability for autonomous systems. Accurate localization is essential for all other mobile robotic tasks, such as path planning and obstacle avoidance. GNSS-based outdoor localization is the go-to solution that can even operate without a prior map. GNSS, however, may not always be available.

Manuscript received: November 13, 2023; Revised: January 22, 2024; Accepted: February 13, 2024. This paper was recommended for publication by Editor Javier Civera upon evaluation of the Associate Editor and Reviewers' comments.

The European Commission has supported this work as part of H2020 under grant number 871704 and it has partially been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy, EXC-2070 – 390732324 – PhenoRob.

[1]I. Hroob and M. Hanheide are within the Lincoln Centre for Autonomous Systems (LCAS), University of Lincoln, UK.

[2]B. Mersch and C. Stachniss are with the Center for Robotics, University of Bonn, Germany. Cyrill Stachniss is additionally with the Department of Engineering Science at the University of Oxford, UK, and with the Lamarr Institute for Machine Learning and Artificial Intelligence, Germany.

⋆Authors with equal contributions.

Digital Object Identifier (DOI): see top of this page.

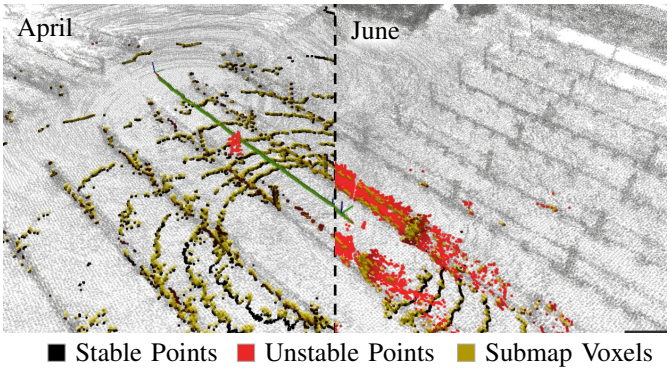

Fig. 1: Our method segments stable and unstable points in 3D LiDAR scans exploiting the discrepancy of scan voxels and overlapping map voxels (highlighted as submap voxels). We showcase two LiDAR scans captured during separate localization sessions within an outdoor vineyard. The scan on the left depicts the vineyard state in April, while the scan on the right reveals environmental changes in plant growth in June.

Map-based mobile robot localization utilizes onboard sensors [25], [41] (i.e. cameras, LiDARs) for vehicle pose estimation by matching the sensory data with the robot's prior belief (i.e. map) about the environment. In this work, we focus on LiDAR-based systems due to the robustness of LiDAR sensors against illumination changes and their ability to measure distances robustly. A common method for LiDAR-based pose estimation is scan matching algorithms [3], [5]. However, due to changes in the environment, scan matching systems may fail since some scan points may not have correspondences in the map, thus leading to failure in accurate estimation of the vehicle pose.

One possibility to improve scan matching localization performance in changing environments is to segment the LiDAR scan into stable and unstable points and to utilize the stable points only for localization. Stable points typically represent elements of a stable object, such as walls, poles, light posts, and tree trunks. To isolate these points, one approach is to employ hand-crafted features for segmentation based on the shape of the object [29], [30]. However, such methods' robustness can be compromised by the varying density of point cloud data, leading to errors in stable points segmentation, thus causing a failure in the localization.

An alternative is to employ deep learning approaches to learn to segment stable points from LiDAR scans, as seen in [13], [21], [33]. Despite the potential of deep learning, these methods often demand substantial amounts of manually annotated data and frequently struggle to generalize effectively

to new, unseen data. To address the issue of labeling, LTS-NET [15] implicitly learns the inherent stable structure in the environment in a self-supervised fashion and utilizes this structure as a landmark to improve vehicle localization in changing environments. The self-supervised training avoids an expensive manual labeling process, however, the generalization capabilities in novel scenes is poor.

In this work, we explore the challenge of enhancing the generalization capabilities of a 3D LiDAR-based learned filter to improve vehicle pose estimation in previously unseen environments. The aim is to enhance the system's adaptability to novel environments that were not encountered during the training of the filter. We propose a generalizable stable points segmentation learned filter by exploiting the discrepancy between the scan and a prior map. For the network part, we utilize sparse convolutions [8] due to their efficiency. To this end, we segment the scan point cloud data into two categories, stable and unstable points, see Fig. 1. In this example, the stable points are the points that belong to long-term stable objects like buildings and poles, while the unstable points are the points that belong to both moving objects in the current scene like walking humans, and objects that tend to change over time such as plant vegetation.

In contrast to the task of moving object segmentation [20], stable points segmentation in 3D LiDAR scan data can segment both present dynamic entities and stationary objects that may undergo positional or perceptual alterations in subsequent instances. Furthermore, unlike supervised semantic segmentation, our method does not require a complex class annotation to supervise the learning, our method can be trained in a self-supervised manner with no manual annotation by leveraging previous observations.

The main contribution of this paper is a novel real-time approach for segmenting stable points from a 3D LiDAR scan. Using these stable points for localization, our approach can enhance scan-to-map matching in changing environments. We achieve this by training a 4D sparse convolutional neural network in a self-supervised manner, allowing it to predict spatio-temporal features in the current scan through the exploitation of discrepancies between the scan and the map data.

In sum, we make two key claims: Our approach is able to (i) segment scan points into stable and unstable points, and utilize the stable points to increase the accuracy of robot long-term localization, (ii) generalize across diverse and unseen environments including settings not encountered during training, leading to improved localization performance, while suitable for online operation on a mobile robot. These claims are backed up by the paper and our experimental evaluation.

II. Related Work

When dealing with localization based on a given map, a common distinction is made between local and global localization. In the latter, the goal is to determine the robot's pose in a map with no prior pose information available. In local localization, i.e. pose tracking, the robot starts from a known pose and it is updated as time progresses. In this work, we address local localization using 3D LiDAR data in a changing environment, commonly known as long-term localization.

LiDAR Map-Based Localization – Two popular approaches for robot LiDAR localization in a pre-built map are probabilistic methods and feature-based methods. Examples of probabilistic methods include Kalman filters [34] and Monte Carlo localization (MCL) [11], which are widely used for robot localization [1], [7], [18]. For instance, Chen et al. [7] utilize MCL to estimate the vehicle's local and global pose within a pre-built mesh-based map representation utilizing range images derived from 3D LiDAR data. On the other hand, feature-based methods such as scan matching techniques [3], [5] estimate the robot pose by aligning the current sensor readings (raw laser scans or visual features) with a pre-built map [17], [22], [28]. Some techniques even combine grids and features [39]. In contrast to the probabilistic methods, scan-matching methods need a good guess of the robot's initial pose. However, they often estimate a smoother trajectory. Therefore, we focus on scan-matching systems and aim to improve their performance in dynamic environments.

Dynamic Environment Localization – In terms of long-term localization, the environment may gradually or suddenly change over time which may impact the accuracy of the robot's pose estimation for scan-matching algorithms [10]. Therefore, researchers explore incorporating new information into the map. For instance, Biber and Duckett [4] introduce an adaptive map that is continuously updated over time. Walcott et al. [38] embed time into mapping to sustain an accurate map in dynamic environments by removing inactive scans and adding new scans. Others exploit sequence information [37] or propose long-term localization approaches [31], [36] that involve temporal mapping. If the current observations do not align with the static map resulting in a failure in pose tracking, a temporary map is generated. This temporary map is later fused with the static map to be utilized in subsequent localization runs. Contrary to the above, our approach does not require a complex map update process, since it uses the initial environment map.

Deep Learning in Long-Term Localization – Several recent methods exploit deep neural networks and semantic information for long-term localization. For example, Tinchev et al. [35] propose a learning-based method for segment matching of trees and localization in diverse environments. At the same time, Kim et al. [16] present a long-term localization method based on a point cloud descriptor called ScanContext, which utilizes a convolutional neural network for localization on a grid map. Lately, Zimmerman et al. [43] overcome the discrepancy between the sensory data and the static map by leveraging human-readable cues. While those methods avoid map updates and work on the sensory data, they are either bound to indoor environments, rely on handcrafted features, or need supervised training. In contrast, our method learns the features in a self-supervised fashion.

Compared to previous works, we propose a novel self-supervised stable points segmentation method by exploiting 4D sparse convolutions. Our method can improve long-term localization performance by using stable points for localization. The use of sparse representations allows us to achieve stable points segmentation in real-time.

III. Our Approach

In this paper, we propose a generalizable stable points segmentation filter to increase the robustness of pose estimation for scan-matching algorithms in changing environments. The pipeline is illustrated in Fig. 2. To this end, we first transform the scan into the global map frame using an initial pose estimate (detailed in Sec. III-A). Then we employ 4D sparse convolutions in Sec. III-B across scan and submap voxels to exploit their discrepancy and increase the network's generalization capability as outlined in Sec. III-C. The 4D sparse CNN is trained in a self-supervised manner, leveraging prior environmental observations to generate long-term stability training labels, as described in Sec. III-D.

A. Map-based 3D LiDAR Localization

When estimating the robot's pose $\boldsymbol{x}_t$ in a given map $\mathcal{M}$ and sensor reading $\mathbf{z}_t$ at time t, the most used localization algorithm is Monte Carlo localization [11]. It can achieve both, local and global localization Alternatively, scan-matching algorithms such as iterative closest point [3] or normal distributions transform [5] can achieve accurate robot localization within a known map. However, unlike Monte Carlo localization they can estimate smoother robot trajectories, but require strong guess of the initial pose and are less robust [1]. In this work, we focus on scan-matching algorithms. We proceed with the assumption of an initial estimate being available. This assumption holds for many robotics applications targeted repeated missions such as data recording and site inspection missions, where the mobile robot typically starts its operations from a fixed initial pose.

The concepts presented in this work are versatile and applicable to both ICP and NDT. However, for this study, we chose NDT due to its efficiency and robustness compared to ICP [19]. To facilitate this, we utilized the NDT localization framework [17], which performs unscented Kalman filter-based pose estimation [34]. The estimated pose provides a strong initial estimate for NDT during scan registration. The sensor transformation matrix to be estimated at time t is defined as follows:

$$\boldsymbol{T}_t = \begin{bmatrix} \boldsymbol{R}_t & \boldsymbol{t}_t \\ \mathbf{0} & 1 \end{bmatrix} \tag{1}$$

where $\boldsymbol{t}_t$ is the position and $\boldsymbol{R}_t$ is the rotation matrix of the sensor with respect to the point cloud map $\mathcal{M}$. NDT aims to find $\boldsymbol{T}_t$ of the current scan $\mathcal{S}_t$ that maximizes the likelihood that $\mathcal{S}_t$ lies on the reference map $\mathcal{M}$ surface. Without loss of generality, we omit the superscript t since all the processes happened at the current time step t. We estimate the transformation matrix $\boldsymbol{T}^*$ as follows:

$$\boldsymbol{T}^* = \underset{\boldsymbol{T}}{\operatorname{argmax}} \sum_i \exp\left(-\frac{d_M^2}{2}\right), \tag{2}$$

where d_M is the Mahalanobis distance

$$d_M = \sqrt{(\boldsymbol{p}_i' - \boldsymbol{\mu}_i)^\mathsf{T} \Sigma_i^{-1} (\boldsymbol{p}_i' - \boldsymbol{\mu}_i)} \tag{3}$$

using the transformed query points $\boldsymbol{p}_i' = T(\boldsymbol{p}_i, \boldsymbol{T})$ obtained by the transformation function T. The expressions Σ_i and $\boldsymbol{\mu}_i$ are the covariance matrix and the mean of the corresponding NDT voxel for the point $\boldsymbol{p}_i'$ looked up in the NDT voxels of the map $\mathcal{M}$.

In a non-static environment, the LiDAR scan measurements are taken from both stable and unstable objects, expressed as:

$$\mathcal{S} = \mathcal{S}_s \cup \mathcal{S}_u. \tag{4}$$

Here, $\mathcal{S}_s$ denotes the subset of points that are measured from stable objects, while $\mathcal{S}_u$ encompasses the points associated with unstable objects. Unstable points are characterized by their lack of corresponding points in the map. The process of filtering out these unstable points serves to enhance the accuracy of scan-matching algorithms by improving the data association between the current scan and map points. We explain the data processing and segmentation steps in the remainder of this section.

B. 4D Sparse Convolution Neural Network

Sparse convolutions are designed to handle sparse data structures efficiently. In the context of 3D point clouds, this is crucial because most of the space in a 3D environment is empty, and processing all empty voxels can be computationally expensive. Several network architectures were proposed to work directly on point cloud data such as PointNet [26], PointNet++ [27], KPConv [32] and PointNetLK [2]. However, most architectures are computationally expensive and can not generalize well for high dimensional spaces [8]. Therefore, for this work, we exploit Minkowski networks [8] since they are memory efficient and fast.

The input to the network consists of a sparse tensor, a representation that efficiently encodes the sparse nature of the point cloud, comprising both the point coordinates $\mathcal{C}$ and the corresponding features $\mathcal{F}$. The sparse tensor is formulated in the following manner, where each entry corresponds to a voxel:

$$\mathcal{C} = \begin{bmatrix} b_1 & x_1 & y_1 & z_1 & t_1 \\ & & \vdots & & \\ b_N & x_N & y_N & z_N & t_N \end{bmatrix} \qquad \mathcal{F} = \begin{bmatrix} \boldsymbol{f}_1^\mathsf{T} \\ \vdots \\ \boldsymbol{f}_N^\mathsf{T} \end{bmatrix}, \tag{5}$$

where b_i is the batch index, t_i is the time index of the 4D tensor and $\boldsymbol{f}_i$ is the feature vector associated to the i-th coordinate voxel. We follow 4DMOS and use a constant feature $\boldsymbol{f}_i = 0.5$ such that the spatio-temporal information is extracted only from the non-empty voxels represented by the 4D coordinates.

C. Generalizable Scan-based Stable Points Segmentation

Enhancing the generalization ability of a network to segment points from 3D LiDAR frames in unfamiliar environments typically involves resource-intensive methods such as expanding the training dataset, regularization, reducing the network size, and employing data augmentation techniques. However, these approaches come with significant costs. Our approach solely uses the spatio-temporal discrepancy between the LiDAR frame and the point cloud map to decide which points are stable, therefore enabling a generalizable setup that does not rely on additional information such as semantics.

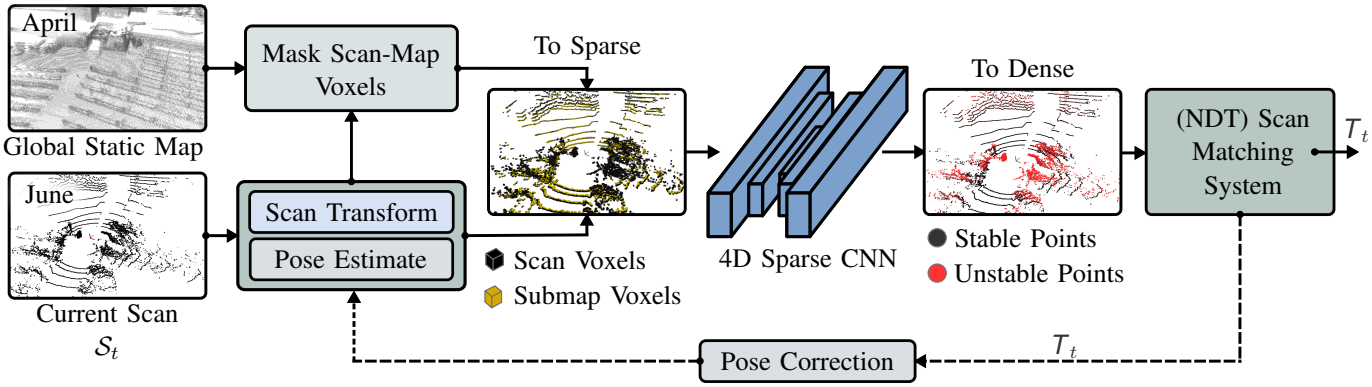

Fig. 2: An overview of our method: Initially, we transform the scan using an initial pose estimate. Next, we voxelize both scan and map points and extract overlapping map voxels, i.e. submap voxels. Both the scan and map voxels are represented as a 4D sparse tensor, with the fourth dimension denoting time t. We then apply sparse 4D convolutions on a joint sparse voxel grid that encompasses both the scan and submap points, leading to the prediction of long-term stability scores for the scan points.

To find the discrepancy between the scan and map point, we first start by transforming the LiDAR scan $\mathcal{S} = \{\boldsymbol{p}_i\}_{i=0}^{N-1}$ into the global map frame utilizing the initial pose prediction from the unscented Kalman filter T', resulting in $\mathcal{S}'$. Then we add a timestamp t to the scan and map points to form a 4D tensor with each point represented as $\boldsymbol{p}_i = [x_i, y_i, z_i, t_i]^\mathsf{T}$, where we use a fixed time t_m for the map points and a fixed time t_s for the transformed scan points $\mathcal{S}'$. The motivation behind this is mainly to distinguish between scan and map points that are falling in the same voxels at later steps. Subsequently, we discretize the scan and map 4D tensors into sparse 4D tensors, utilizing a predefined spatial resolution. The scan and map sparse voxel grid coordinates are denoted as follows: $\mathcal{C}_S \in \mathbb{R}^{4\times n}$ and $\mathcal{C}_M \in \mathbb{R}^{4\times m}$. Here, n and m are the numbers of scan and map voxels, respectively. Each voxel coordinate is represented using its central Cartesian position. It is important to note that the original point coordinates are preserved within their respective voxels to recover a per-point segmentation.

Next, we merge the scan and map 4D tensors into a unified tensor. Given that scan and map voxels share the same coordinate frame, this merging process highlights the discrepancies between the scan and map voxels, revealing three possible scenarios for a voxel. Firstly, if a voxel encompasses both scan and map timestamps, it suggests association with a stable object. Secondly, if a spatial voxel exclusively contains the scan timestamp, it indicates a potential association with an unstable object. Lastly, if a voxel solely possesses the map timestamp, it signifies that the voxel is either beyond the scan range or has been obscured by another object.

To estimate the stability confidence score for each point, we employ a sparse CNN designed for stability inference through regression. This involves applying sparse convolution to the unified scan and map sparse 4D tensor. Our sparse CNN is derived from the modified MinkUNet14 [8], initially introduced in 4DMOS [20]. We repurpose this network as a regression model, with a specific modification to the final layer. In this adaptation, we utilize the sigmoid function to predict confidence scores for stable points, ensuring the values range between 0 and 1 for each point.

Passing the complete unified 4D tensor to the network could adversely affect the network's inference time performance, primarily due to the substantial size of the unified tensor, driven by the dimension of the map tensor. To tackle this challenge and enhance inference speed, we opt to prune the unified sparse tensor. Specifically, we eliminate sparse voxels exclusively containing the map timestamp only. We keep only the voxels that contain at least the scan timestamp as illustrated in Fig. 3. This decision stems from our specific interest in inferring stability confidence scores only for the scan points.

Finally, we segment the stable points $\mathcal{S}_s$ from the current scan $\mathcal{S}$ based on the predicted stability confidence scores assigned to $\mathcal{S}$, where we apply a fixed threshold ϵ for segmenting the stable points. Unlike our prior work [15], our approach leverages this scan-map discrepancy effectively in segmenting stable points, contributing significantly to the network's robust generalization performance.

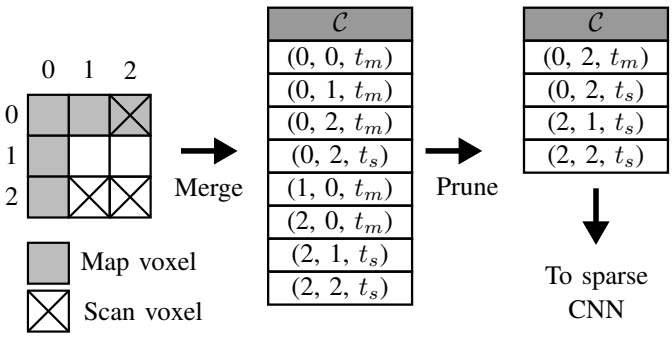

Fig. 3: Our method prunes map voxels without scan correspondence, significantly improving performance by eliminating unnecessary voxels. In this example, the map depicts a wall corner, and the illustrated scan voxels reveal the occlusion of some map voxels caused by an obstacle. We show the resulting voxel coordinates $\mathcal{C}$ before and after pruning and indicate the map and scan timestamps as t_m and t_s, respectively.

D. Training Labels for Sparse CNN

We generate the training labels for the sparse CNN in an unsupervised fashion based on prior work [15], mainly to avoid the manual labeling process as it is time-consuming. A point is considered stable if we have at least two observations of its environment. We first build point cloud maps of the observations denoted as $\{\mathcal{M}\}_0^k$ using a simultaneous localization and mapping system such as FAST-LIO [40], where k is the number of observations of the environment, along with their associated occupancy grid OctoMaps [14].

Assuming the point cloud maps are roughly aligned, we fine-align them using ICP. The labeling procedure starts with selecting a reference map $\mathcal{M}_i$ to be labeled. For each point $\boldsymbol{p} \in \mathcal{M}_i$, a spatio-temporal stability label is assigned based on the maximum spatial distance d to the nearest point in all other maps while accounting for occlusions by querying the occupancy of the point location in the query map.

This distance value is transformed into a unitless value using the cumulative distribution function of an exponential function: $F(d) = 1 - \exp(-d)$. This transformation bounds the continuous value between 0 and 1, effectively representing long-term spatio-temporal stability, where a value close to 0 suggests a stable point, whereas a value approaching 1 signifies an unstable point. For an in-depth understanding of this labeling pipeline and our approach to handle occlusions, we direct the interested reader to our prior work [15].

IV. Experimental Evaluation

The main focus of this work is to segment unstable points from LiDAR scans and utilize the remaining stable points to improve the localization performance of scan-matching algorithms in changing environments. We present our experiments to show the capabilities of our method. The results of our experiments also support our key claims, which are: (i) increasing the robustness of robot long-term localization, (ii) generalizing well to different environments without model retraining.

A. Experimental Setup

1) Datasets: We demonstrate our method's effectiveness in learning to segment stable points and improving long-term localization using the Bacchus long-term (BLT) dataset [23]. This dataset was collected in semi-structured agricultural environments over several months. Additionally, we assess our approach's generalization by employing two more datasets. One is Riseholme, which is a vineyard at Riseholme campus and also part of BLT. The other dataset is a parking lot from the north campus long-term (NCLT) dataset [6], which has diverse objects not found in BLT, thus challenging our model's transfer capabilities.

2) Baselines: We compared our approach to four baseline methods: (i) raw, which uses the unfiltered scans for localization, (ii) mask, which utilizes the masked submap voxels that are associated with scan voxels, (iii) 4DMOS [20], which is a method that filters dynamic objects in a sequence of past scans not considering the prior map and therefore the points' long-term stability, and (iv) LTS-NET [15], which filters unstable points based on long-term stability labels but does not leverage the scan-map discrepancy.

3) Implementation Details: We set the quantization size for the sparse voxel grid to $0.1\,\mathrm{m}$ mainly to not lose details of the features. We train our 4D sparse CNN in a self-supervised manner by using the auto-generated stability labels Sec. III-D, as a cost function we use the root mean square error (RMSE) $\mathcal{L}$ loss, and we supervise the training on scan data only. For generating the training data, we utilize the BLT dataset. Specifically, we use sequences from April 20th and June 1st for training, while sequences from June 8th are used for validation. The model undergoes training for 60 epochs, and we save the best-performing model based on its performance on the validation dataset. Additionally, we augment the training batches by applying random flipping, rotating, and scaling, as outlined in [20]. After predicting long-term stability confidence score with our method, we use a fixed threshold of $\epsilon = 0.84$ for filtering stable points which is chosen based on the localization performance on the training and validation set.

Both the presented approach and LTS-NET were trained on the automated data generated from the BLT dataset, while we did not train 4DMOS on this dataset for two reasons: (i) 4DMOS claims to generalize well in new unseen environments, (ii) the auto-generated labels from the BLT dataset will not work with 4DMOS since the labels do not indicate if the object is currently moving.

B. Localization Performance in Agricultural Environments

In this section, we conduct experiments to back up our first claim and evaluate the ability of our method to localize in changing environments using the segmented stable stable points only.

To evaluate the performance of improving the accuracy of scan-to-map localization in changing environments through the utilization of stable scan points, we first build a base map using earlier sequences of the BLT dataset. Particularly, we use the April 6th sequence of an early growth stage therefore containing only stable objects. Subsequently, we utilize the data from the later sessions to perform localization within the base map. For the quantitative evaluation, we use the RMSE of the absolute trajectory error (ATE) [42]. However, it is important to note that the localization algorithm might fail to estimate a reliable pose and start providing inaccurate poses. To account for this, we consider the localization as failed if the ATE exceeds a specific threshold τ ($\tau = 1.5$ m for our use case), marking the time when the localizer failed. We excluded estimated poses beyond this point from the evaluation. To conduct a fair trajectory evaluation, we employ the trajectory duration ratio metric [9]. It represents the ratio between the duration of the estimated trajectory (est) and the total duration of the ground truth trajectory (gt). Specifically:

$$R_{ts} = \frac{\Delta\ t_{\text{est}}}{\Delta\ t_{\text{gt}}}, \tag{6}$$

where a value closer to $100\,\%$ indicates a more accurate estimation. The localization performance of all methods is summarized in Tab. I.

Method	Raw		Mask		4DMOS		LTS-NET		Ours	
Seq/Metric	$RMSE$	R_{ts}	$RMSE$	R_{ts}	$RMSE$	R_{ts}	$RMSE$	R_{ts}	$RMSE$	R_{ts}
April-20th*	0.133±0	**100±0**	0.385±0.03	64.8±12	0.132±0	**100±0**	**0.093±0**	**100±0**	0.128±0	**100±0**
June-1st*	0.352±0	72±0	0.777±0.10	8.8±2	0.546±0	**100±0**	0.3±0	**100±0**	**0.288±0**	**100±0**
June-8th**	0.382±0	**100±0**	0.697±0.08	8.2±4	0.366±0	**100±0**	**0.272±0**	**100±0**	0.281±0	**100±0**
June-29th	0.483±0	8.8±0	0.610±0.01	8.2±1	0.451±0	8.8±0	**0.469±0**	**100±0**	0.528±0.02	90.7±0
July-13th	0.281±0.03	75±0	0.663±0.02	12.2±0.04	**0.201±0**	73.7±0	0.416±0	11.1±0	0.350±0.06	**87±0**

TABLE I: Averaged localization performance comparison between baseline methods and our proposed approach ('Ours') from five experiments, including standard deviations. RMSE results are reported in meters, and R_{ts} values are expressed in percentage. The * indicates the training sequences of the stable points segmentation and the ** indicates the validation sequence.

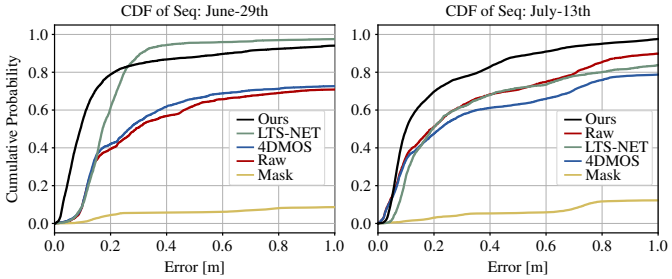

Fig. 4: Plots of the cumulative distribution function of the translational localization error for the BLT sequences June-29th and July-13th.

In the initial vineyard stages on April 20th, when the environment was relatively stable, most methods exhibit similar localization performance, with a slight advantage for LTS-NET. However, the mask baseline fails when the robot rotated at the end of a row. Subsequent sessions show degradation and eventual failure in the localization performance of both raw scans and 4DMOS-segmented scans, particularly in the June 29th sequence. This is due to 4DMOS's limitation in segmenting only dynamic objects, such as pedestrians, while neglecting unstable objects like overgrown vegetation. These unstable points will not have a map correspondence as illustrated in Fig. 1, thus causing scan matching failure. Conversely, our method and LTS-NET consistently deliver competitive performance across various sequences and metrics. An interesting observation is the results for the July 13th session, where LTS-NET initially tracks only about 11% of the trajectory, while our method exhibits a more robust performance, tracking 87% of the entire trajectory. In addition to RMSE and R_{ts}, we visualize the empirical cumulative distribution function [24] to evaluate the robustness of the system and to assess the registration accuracy between the base map and the LiDAR scan. We show the cumulative distribution function plots for the test sequences in Fig. 4. The closer the curve is towards the upper left corner, the smaller are the expected errors and the more robust is the system. The results verify that the proposed approach is more robust compared to the baseline.

The mask baseline consistently fails in all sessions. The reason behind this failure is that this baseline relies on the accuracy of the estimated scan pose to segment the true associated map voxels; thus a misalignment between the scan and the map larger than the size of the voxelization can result in a failure to segment the true associated map points, thus causing a failure in the localization.

C. Generalization Capabilities

Dataset/Method	Raw	4DMOS	LTS-NET	Ours
Riseholme	0.264	0.263	0.290	**0.261**
NCLT-115	0.165	0.166	0.165	**0.157**
NCLT-202	0.163	0.160	0.167	**0.157**
NCLT-219	0.170	0.164	0.158	**0.156**

TABLE II: Generalization performance of the proposed method compared to the baselines in new environments. We report the RMSE of the estimated trajectory in meters.

Next, we assess our second claim about the proposed method's generalization capability to segment the stable points and to enhance long-term localization in new and diverse environments. We conducted experiments in two environments. The first environment is a vineyard located at the Riseholme campus of the University of Lincoln, while the second setup is a parking lot from the NCLT dataset. In both cases, we do not retrain the models or use any domain adaptation techniques. Further, we employ a base map representing the static structure. In the vineyard, we observe changes due to plant growth, while the parking lot poses two distinct challenges: alterations in the parking lot shape based on the number of cars and the presence of plant vegetation as well as moving objects in the sequence. For the NCLT dataset, we use data from the sequences 2012-01-15, 2012-02-02, and 2012-02-19, denoted as NCLT-115, NCLT-202, and NCLT-219, respectively.

Tab. II summarizes the method's localization performance compared to the baseline. The reported results are the averages of five runs. The deviation of the runs is not presented since the results were consistent. Additionally, we do not report the R_{ts} metric and only report the RMSE of the ATE since the localizer effectively tracks the robot throughout the entire trajectory for all methods. We exclude the mask baseline from these experiments as it consistently fails in all trials due to initial pose misalignment.

The results in Tab. II confirm the validity of our second claim. Notably, the localization performance of raw scans and segmented scans from 4DMOS exhibit similarities, suggesting that dynamic objects such as moving pedestrians or cars have minimal impact on the localization performance. To back this up, we manually labeled the dynamic objects and found their proportion to be 0.73 % of all points in the three sequences and 4.22 % of the points belonging to movable objects like pedestrians and cars. This indicates that the majority of movable and therefore unstable points is not dynamic. We hypothesize that the utilization of stable points significantly influence the localization performance. Furthermore, the uti-

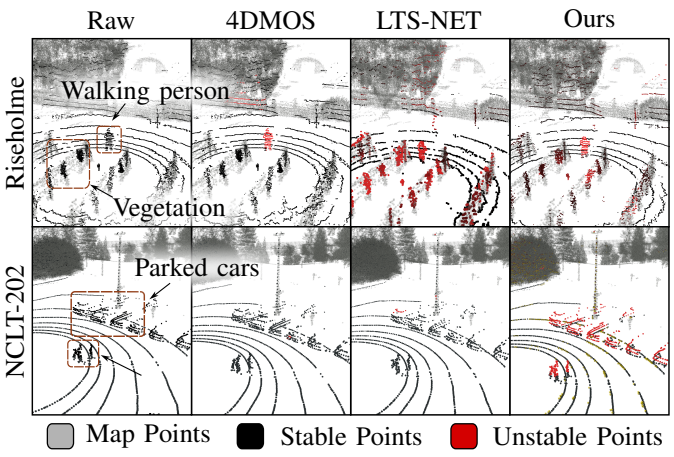

Fig. 5: Comparison of the generalization ability of our method to multiple baselines. Riseholme is recorded in an agricultural environment with walking persons and changing vegetation whereas NCLT is a parking lot with parked cars.

lization of both scan and submap voxels as an indication of discrepancy enhances the system's generalization capability, making it more capable at segmenting and utilizing stable points in new and diverse environments as shown in Fig. 5. The proposed method successfully segments unstable elements in the Riseholme dataset, including humans and vegetation, while in the NCLT dataset, it accurately identifies parking cars and pedestrians as unstable objects, a task where both 4DMOS and LTS-NET fail. Possible reasons are that 4DMOS segments stable points based on their current motion and LTS-NET based on object shapes it has seen during training which is an agricultural environment in this experiment.

D. Filtering of Unstable Points

The previous experiments suggest that our approach successfully localizes in changing environments by segmenting and filtering unstable points. To give a more detailed reasoning why our system improves upon existing methods, we provide a quantitative evaluation of the classification performance of unstable points. We report standard metrics such as intersection-over-union (IoU), precision, recall, and F1 score [12] for unstable points on the validation and test set of BLT and NCLT. We focus on the evaluation of unstable points since their removal is more critical than for example keeping all static points. It is important to note that we use the auto-generated labels from Sec. III-D for evaluation, as no ground truth labels are available. The results in Tab. III and Tab. IV illustrate the system's effectiveness in segmenting unstable points in contrast to the baselines. Our approach achieves the highest recall and therefore removes more unstable points than the baselines resulting in a better localization in Sec. IV-B and Sec. IV-C. Additionally, the results in Tab. IV again confirm the ability of our proposed method to segment unstable points across novel and unseen environments. Note that the performance gap for 4DMOS is due to the fact that 4DMOS segments moving objects only which is only a subset of the moving points.

Seq	Method	IoU	Precision	Recall	F1
June-8th**	**4DMOS**	0.039	0.554	0.039	0.072
	LTS-NET	0.643	0.836	0.738	0.779
	Ours	**0.727**	**0.861**	**0.827**	**0.839**
June-29th	**4DMOS**	0.065	0.47	0.068	0.105
	LTS-NET	0.637	0.878	0.701	0.775
	Ours	**0.784**	**0.924**	**0.836**	**0.877**
July-13th	**4DMOS**	0.006	0.541	0.006	0.012
	LTS-NET	0.611	0.846	0.687	0.755
	Ours	**0.78**	**0.875**	**0.881**	**0.875**

TABLE III: Segmentation performance of unstable points for the validation and test sequences of the BLT dataset. We report the average over all scans in the corresponding sequence. All metrics are computed for the unstable points. Best results in bold. The ** indicates data used for validating the stable points segmentation.

Seq	Method	IoU	Precision	Recall	F1
115	**4DMOS**	0.113	0.359	0.139	0.174
	LTS-NET	0.054	0.269	0.07	0.099
	Ours	**0.262**	**0.382**	**0.483**	**0.391**
202	**4DMOS**	0.198	0.649	0.23	0.302
	LTS-NET	0.152	0.601	0.167	0.251
	Ours	**0.585**	**0.684**	**0.785**	**0.721**
219	**4DMOS**	0.115	0.446	0.129	0.174
	LTS-NET	0.075	0.277	0.096	0.132
	Ours	**0.502**	**0.616**	**0.717**	**0.638**

TABLE IV: Segmentation performance of unstable points for the NCLT dataset. The reported results are averaged over all scans. All metrics are computed for the unstable points. Best results in bold.

E. Runtime

We summarize the inference time of our method compared to 4DMOS and LTS-NET in Tab. V on an NVIDIA GTX 1080ti GPU. The results demonstrate that the proposed approach can run sufficiently fast for mobile robots. Furthermore, our approach shows a smaller GPU memory demand of 1047 MB compared to 1703 MB for 4DMOS, and 9487 MB for LTS-NET.

Dataset	LiDAR	4DMOS	LTS-NET	Ours
BLT	**16-beams**	0.052 (19.1)	0.095 (10.5)	**0.037 (27.3)**
NCLT	**32-beams**	0.048 (20.7)	0.101 (9.9)	**0.036 (27.8)**

TABLE V: Average inference time for 3D LiDAR frames, the results presented in seconds and (Hz).

V. Conclusion

In this paper, we presented a novel approach to increase the accuracy of scan-to-map-based localization in changing environments. Our approach segments the scan points into stable and unstable points based on their long-term stability, and we use only the stable points for localization. The backbone of our method is a 4D sparse CNN that we train in a self-supervised fashion. Initially, we train and evaluate our method using the BLT dataset, followed by assessing its generalization capabilities in two additional datasets. The outcome indicates an improvement in localization performance, successful generalization to unseen data, and a runtime suitable for mobile robots, which supports all claims made in this paper.

Despite the effectiveness of our proposed approach, we rely on an accurate pose estimate for the initial alignment of the scan with the map to accurately determine discrepancies between scan and map data. This alignment should be a reasonable initial guess to avoid wrong segmentation leading to

a localization failure. In future research, we aim to strengthen the robustness of the initial estimate by incorporating more odometry data.

REFERENCES

[1] N. Akai. Efficient Solution to 3D-LiDAR-based Monte Carlo Localization with Fusion of Measurement Model Optimization via Importance Sampling. *arXiv preprint*, arXiv:2303.00216, 2023.

[2] Y. Aoki, H. Goforth, A.S. Rangaprasad, and S. Lucey. PointNetLK: Robust & Efficient Point Cloud Registration Using PointNet. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[3] P. Besl and N. McKay. A Method for Registration of 3D Shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 14(2):239–256, 1992.

[4] P. Biber and T. Duckett. Dynamic Maps for Long-Term Operation of Mobile Service Robots. In *Proc. of Robotics: Science and Systems (RSS)*, 2005.

[5] P. Biber and W. Straßer. The normal distributions transform: A new approach to laser scan matching. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2003.

[6] N. Carlevaris-Bianco, A. Ushani, and R. Eustice. University of Michigan North Campus long-term vision and lidar dataset. *Intl. Journal of Robotics Research (IJRR)*, 35(9):1023–1035, 2016.

[7] X. Chen, I. Vizzo, T. Läbe, J. Behley, and C. Stachniss. Range Image-based LiDAR Localization for Autonomous Vehicles. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[8] C. Choy, J. Gwak, and S. Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[9] F. Crocetti, E. Bellocchio, A. Dionigi, S. Felicioni, G. Costante, M.L. Fravolini, and P. Valigi. ARD-VO: Agricultural Robot Data Set of Vineyards and Olive Groves. *Journal of Field Robotics (JFR)*, 40(6):1678–1696, 2023.

[10] A. Das, J. Servos, and S. Waslander. 3D Scan Registration Using the Normal Distributions Transform with Ground Segmentation and Point Cloud Clustering. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2013.

[11] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 1999.

[12] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *Intl. Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010.

[13] H.W. F. Duerr, M. Pfaller and J. Beyerer. LiDAR-based Recurrent 3D Semantic Segmentation with Temporal Memory Alignment. In *Proc. of the Intl. Conf. on 3D Vision (3DV)*, 2020.

[14] A. Hornung, K. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. *Autonomous Robots*, 34(3):189–206, 2013.

[15] I. Hroob, S. Molina, R. Polvara, G. Cielniak, and M. Hanheide. Learned Long-Term Stability Scan Filtering for Robust Robot Localisation in Continuously Changing Environments. In *Proc. of the Europ. Conf. on Mobile Robotics (ECMR)*, 2023.

[16] G. Kim, B. Park, and A. Kim. 1-day learning, 1-year localization: Long-term LiDAR localization using scan context image. *IEEE Robotics and Automation Letters (RA-L)*, 4(2):1948–1955, 2019.

[17] K. Koide, J. Miura, and E. Menegatti. A Portable Three-dimensional LIDAR-based System for Long-term and Wide-area People Behavior Measurement. *Intl. Journal of Advanced Robotic Systems*, 16(2), 2019.

[18] D. Maggio, M. Abate, J. Shi, C. Mario, and L. Carlone. Loc-NeRF: Monte Carlo Localization using Neural Radiance Fields. *arXiv preprint*, arXiv:2209.09050, 2022.

[19] M. Magnusson, H. Andreasson, A. Nuechter, Achim, and J. Lilienthal. Appearance-Based Loop Detection from 3D Laser Data Using the Normal Distributions Transform. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2009.

[20] B. Mersch, X. Chen, I. Vizzo, L. Nunes, J. Behley, and C. Stachniss. Receding Moving Object Segmentation in 3D LiDAR Data Using Sparse 4D Convolutions. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):7503–7510, 2022.

[21] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. RangeNet++: Fast and Accurate LiDAR Semantic Segmentation. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[22] N. Naikal, J. Kua, G. Chen, and A. Zakhor. Image augmented laser scan matching for indoor dead reckoning. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[23] R. Polvara, S. Molina, I. Hroob, A. Papadimitriou, K. Tsiolis, D. Giakoumis, S. Likothanassis, D. Tzovaras, G. Cielniak, and M. Hanheide. Bacchus Long-Term (BLT) Data Set: Acquisition of the Agricultural Multimodal BLT Data Set with Automated Robot Deployment. *Journal of Field Robotics (JFR)*, 40(8), 2023.

[24] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. Comparing icp variants on real-world data sets: Open-source library and experimental protocol. *Autonomous Robots*, 34:133–148, 2013.

[25] H. Porav, W. Maddern, and P. Newman. Adversarial Training for Adverse Conditions: Robust Metric Localisation Using Appearance Transfer. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.

[26] C.R. Qi, H. Su, K. Mo, and L.J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[27] C. Qi, K. Yi, H. Su, and L.J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2017.

[28] J. Roewekaemper, C. Sprunk, G. Tipaldi, C. Stachniss, P. Pfaff, and W. Burgard. On the Position Accuracy of Mobile Robot Localization based on Particle Filters combined with Scan Matching. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2012.

[29] A. Schaefer, D. Büscher, J. Vertens, L. Luft, and W. Burgard. Long-term urban vehicle localization using pole landmarks extracted from 3-D lidar scans. In *Proc. of the Europ. Conf. on Mobile Robotics (ECMR)*, 2019.

[30] R. Spangenberg, D. Goehring, and R. Rojas. Pole-Based Localization for Autonomous Vehicles in Urban Scenarios. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[31] C. Stachniss and W. Burgard. Mobile Robot Mapping and Localization in Non-Static Environments. In *Proc. of the National Conf. on Artificial Intelligence (AAAI)*, 2005.

[32] H. Thomas, C. Qi, J. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas. KPConv: Flexible and Deformable Convolution for Point Clouds. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2019.

[33] H. Thomas, B. Agro, M. Gridseth, J. Zhang, and T.D. Barfoot. Self-Supervised Learning of Lidar Segmentation for Autonomous Indoor Navigation. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[34] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[35] G. Tinchev, A. Penate-Sanchez, and M. Fallon. Learning to see the wood for the trees: Deep laser localization in urban and natural environments on a CPU. *IEEE Robotics and Automation Letters (RA-L)*, 4(2):1327–1334, 2019.

[36] G.D. Tipaldi, D. Meyer-Delius, and W. Burgard. Lifelong localization in changing environments. *Intl. Journal of Robotics Research (IJRR)*, 32(14):1662–1678, 2013.

[37] O. Vysotska and C. Stachniss. Effective Visual Place Recognition Using Multi-Sequence Maps. *IEEE Robotics and Automation Letters (RA-L)*, 4(2):1730–1736, 2019.

[38] A. Walcott-Bryant, M. Kaess, H. Johannsson, and J.J. Leonard. Dynamic Pose Graph SLAM: Long-term Mapping in Low Dynamic Environments. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2012.

[39] K. Wurm, C. Stachniss, and G. Grisetti. Bridging the gap between feature- and grid-based slam. *Journal on Robotics and Autonomous Systems (RAS)*, 58(2):140 – 148, 2010.

[40] W. Xu and F. Zhang. FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter. *IEEE Robotics and Automation Letters (RA-L)*, 6(2):3317–3324, 2021.

[41] H. Yin, Y. Wang, L. Tang, X. Ding, S. Huang, and R. Xiong. 3D LiDAR Map Compression for Efficient Localization on Resource Constrained Vehicles. *IEEE Trans. on Intelligent Transportation Systems (TITS)*, 22(2):837–852, 2020.

[42] C. Zhang, M.H. Ang, and D. Rus. Robust lidar localization for autonomous driving in rain. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[43] N. Zimmerman, L. Wiesmann, T. Guadagnino, T. Läbe, J. Behley, and C. Stachniss. Robust Onboard Localization in Changing Environments Exploiting Text Spotting. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2022.