IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

ProbPer-LiLo: Probabilistic Persistency Modeling for Life-Long Mapping

Waqas Ali[1], Yixi Cai[1], Patric Jensfelt[1] and Thien-Minh Nguyen[2]

***Abstract*—3D mapping is vital for a broad range of applications that rely on a consistent and accurate representation of the environment. Change is an ever-persistent force in our world and with the evolution of a scene its 3D map becomes outdated. Thus, a mapping framework that can adapt and refine the 3D maps with the changes in the scene is necessary. In this paper, we propose a lifelong mapping framework where map maintenance is based on two objectives including the preservation of static structures and the refinement of the 3D map. To preserve only the static structures, we classify the object's state and remove the dynamic objects and the quasi-static objects, i.e., objects which temporarily appear static. For classifying the state of objects, we propose a discrete probabilistic solution utilizing a factor graph. Using this classification, we generate static maps from multiple sessions which are used for map refinement. The refinement is based on change detection and map update, leveraging semantic and geometric information. For the evaluation, we collect a multi-campus lifelong dataset as an extension of the MCD datasets from KTH and NTU campuses. The proposed approach is capable of accurately detecting quasi-static objects even in highly dynamic environments. Our system demonstrates state of the art performance in large scale environments. Furthermore, our approach can handle both SLAM-generated and survey-grade maps.**

***Index Terms*—Mapping, Probabilistic Inference, SLAM**

I. INTRODUCTION

3D mapping systems serve as a foundational technology for a diverse set of applications. Such applications require 3D maps to provide an accurate and consistent representation of the environment for an extended period of time. However, continual changes in the world pose a significant challenge as structural alterations, changes in vegetation, or demolition of existing structure can make existing maps obsolete. Thus, to utilize a 3D map to its full potential, it must evolve with the environment, necessitating framing mapping as a lifelong process. ***We define lifelong mapping (LLM) as the process of building and maintaining an accurate and consistent representation of the static part of the environment over an extended period of time and we focus on LiDAR based mapping.*** *Map building* is typically performed by SLAM algorithms for robotic applications [1]–[3] but can also be done by merging scans acquired by survey-grade LiDAR [4]. *Maintaining* the map has two objectives. First, we need to preserve the representation of the static structures within a map while removing *non-static* objects. Both dynamic objects as well as *quasi-static objects*, that is, objects that are temporary static such as a parked car should be removed. Quasi-static objects as shown in Figure 1 are challenging because they often remain static in an entire mapping session. Second, we need to detect changes to the static structures and refine the 3D map over time as more observations are available.

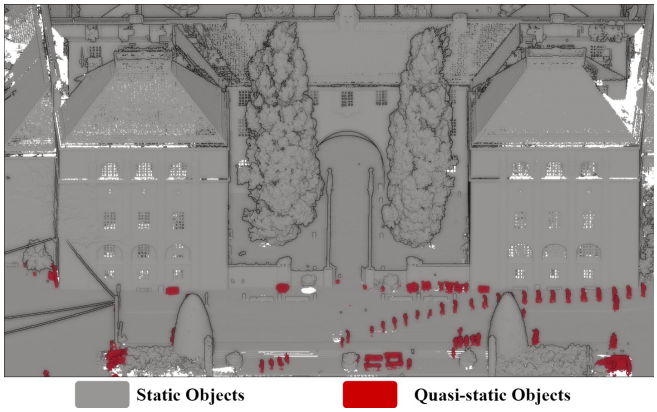


Fig. 1: The map from KTH campus with static and quasi-static objects. The consistent representation can be provided by only preserving static structures in the map.

There are several works that focus on visual lifelong mapping [5]–[7]. Nevertheless, LiDAR-based lifelong mapping has received less attention. However, advances in LiDAR-based mapping have enabled the creation of high-quality maps with cm-level accuracy. Such highly precise 3D maps are essential for robust autonomous navigation [8], UAV-based infrastructure inspection [9], [10] and large scale environmental exploration [11]. Beyond robotic applications, LiDAR-based 3D maps can be vital for environmental change detection, optimal route planning, climate impact assessment, historical preservation, and ancillary studies for urban planning. Recently, a few LiDAR-based LLM frameworks have been proposed [12], [13], which remove dynamic objects from the map and then perform change detection and update. In these methods, changes are defined in a binary manner and the prior information is simply replaced by the current observations, thus discarding rather than fusing information. Furthermore, quasi-static objects are not removed from the map in these methods. The concept of ephemerality (how "temporary" something is) was introduced in [14], where they

Manuscript received: August, 14, 2025; Revised November, 4, 2025; Accepted December, 9, 2025.

This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP).

[1]Waqas Ali, Yixi Cai and Patric Jensfelt are with the Division of Robotics, Perception, and Learning (RPL), KTH Royal Institute of Technology, Stockholm 114 28, Sweden. Corresponding Author: Yixi Cai (yixica@kth.se).

[2]Thien-Minh Nguyen is with the School of Mechanical and Mining Engineering (SMME), The University of Queensland, Brisbane, QLD 4072, Australia.

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

built a lifelong mapping framework based on the ephemerality of the LiDAR points. Another common trait in existing work on lifelong mapping, visual as well as LiDAR, is that they operate in relatively small environments. The reason for this key limitation of earlier methods is their reliance on point-wise analysis, which hinders scalability. As a result these methods become ineffective for large-scale maps, where computational demands grow prohibitively. Hence, we determine that a new LLM method is needed to handle significantly larger environments. In addition, the map maintenance task should be grounded on some probabilistic model to effectively remove quasi-static objects from the map, as well as fusing new observations with prior information, rather than using binary classification and direct replacement. This motivates the development of our method, so-called PropPer-LiLo.

In ProbPer-LiLo, we focus on correctly identifying the truly static objects. This means that the classification border is to be determined so that static objects are on one side, and quasi-static and dynamic objects are on the other, effectively making no distinction between quasi-static and dynamic objects. We draw inspiration from the work of Sanders et al. [15] that proposed a theory of the quasi-static world based on image sequences. Extending this theory to 3D maps, we propose a probabilistic solution leveraging a factor graph to detect quasi-static objects in large scale 3D maps. The factor graph integrates the semantic labels of points with their *temporal consistency* over multiple mapping sessions to accurately classify the objects as static or quasi-static. This allows us to preserve only static objects in the map, which contributes to assessing meaningful changes in the environment over time. Our system can operate with both SLAM maps and survey-grade maps. The main contributions of our work are:

- We propose a novel discrete probabilistic modeling approach for accurate classification of static and quasi-static objects, based on a factor graph formulation, leveraging semantic objects as variables across multi-session maps.
- We propose a comprehensive map change detection and update approach using multi-session maps, that scales to large environments, and that can handle prior maps from SLAM approaches built with consumer level LiDARs as well as very dense maps created by survey-grade LIDARs.
- We demonstrate state-of-the-art performance for change detection and map update tasks with extensive experiments on a novel set of real-world datasets extending the publicly released MCD dataset [16] with data captured two years after the initial dataset, thus enabling real lifelong mapping evaluation.
- The complete framework of the proposed system will be publicly released along with the experimental resources.

II. Related Work

The components of lifelong mapping have been addressed individually in the literature. Regarding the classification of static and non-static objects, dynamic object detection is considered essential for producing consistent maps. A common approach for the dynamic object detection is based on free-space in the occupancy voxel structure of the map [17]. Other methods adopt differential based dynamic points removal leveraging range images [18], occupancy grids [19], [20] and binary-encoded matrix operation [21]. Furthermore, several learning based methods [22], [23] have been proposed to address dynamic point removal. While these method work efficiently on detecting the moving points in the map, they cannot handle quasi-static objects which might appear static but effect the long-term environment representation in the map.

Map refinement concerns aligning multiple sessions, change detection and update of the 3D maps. Several methods [24], [25] focus on an efficient pose-graph structure to merge multi-session sub-maps and generate an accurate global map. Change detection among 3D point cloud maps is also vital for reuse of the maps with time. Stilla et. al. [26] provided an extensive overview of map change detection approaches present in the literature. An issue observed in their analysis is the need for a method that can generalize well to different sensor types used for map generation. They also point out that point cloud semantics can be important for change detection and categorization. Gehrung et. al. [27] proposed change detection using the difference of occupancy grid maps and aggregate the changes into a delta octree. Langer et. al. [28] used semantic information for change detection coupled with geometric verification for vision based reconstruction scenes. Adam et. al. [29] proposed a change detection approach based on geometric transformation consistency. Similarly, Rowell et. al. [30] proposed an object-level change detection approach relying on geometric information. Most of these methods focus on specific cases or limited environments. In this work, **we build a change detection approach that combines semantic and geometric information applicable for large scale maps.**

Rosen et al. [31] proposed a feature persistence model based on Bayesian survival analysis and built a recursive Bayesian filter based on this model to estimate persitence of feature in an environment at any given time. Nobre et al. [32] proposed a joint formulation of the feature persistence by learning the underlying structure of the environment. This joint belief estimate is then integrated into the data association for graph SLAM. Deng et. al. [33] fuse the feature persistence model to visual SLAM method for detecting dynamic visual features. Perpetua [34] models the semi-static features by incorporating persistence and emergence filters into the mixture model. The states of features in an environment are estimated through prior knowledge and learning the dynamics through mixture model. In this work, we used the feature persistence model to build observation factor for building factors. The inference results in a robust and accurate estimate for the feature states.

A few structural lifelong mapping approaches have been proposed that perform step-by-step static map classification and map refinement. A room-scale map maintenance framework was proposed by Ambrus et. al. [35] using an RGB-D camera and removing dynamic objects based on temporal persistence. In addition, static structures are reconstructed and preserved for consistent map representation. A modular lifelong mapping framework LT-mapper [12] was proposed by Kim et. al., which performs multi-session map alignment, dynamic object removal and change detection. Inter-session loop detection is used for alignment and Removert [18] is

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

applied for dynamic object removal. Their map alignment can suffer for wrong loop closure candidates and change detection is based on disappeared or appeared points (i.e., binary classification), which can result in inconsistent maps. Yang et. al. [13] proposed a map version control method with lifelong mapping based on map alignment, dynamic object removal and change detection. Their system also treats change detection as a binary problem. Gil et. al. proposed ELite [14], which performs lifelong mapping based on point-wise ephemerality to deal with transient objects. Bayesian inference is used for computing global ephemerality which is applied for change detection and update. However, point-wise ephemerality can result in removal of parts of static structures. To counter these challenges, **we propose a complete lifelong mapping framework with a probabilistic model for map refinements by preserving static structures and a voxel based change detection and update step.**

III. PRELIMINARIES

A. Notations

We consider a single mapping session as a collection of scans $\{z_1, z_2, ..., z_k\}$ along with respective poses $\{\mathbf{T}_1^w, \mathbf{T}_2^w, ..., \mathbf{T}_k^w\}$ where $\mathbf{T}_i^w \in \mathrm{SE}(3)$ and w is the world frame. The scans and associated poses are used to construct the map denoted by $\mathcal{M}$. The map $\mathcal{M}$ is a set of points denoted as $\{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, ..., \mathbf{f}_n\}$ where $\mathbf{f}_i \in \mathbb{R}^3$, with associated semantic labels $\{l_1, l_2, l_3, ..., l_n\}$. For persistence modeling, we denote the state of the feature $\mathbf{f}_i$ with variable $X_i \in \{0, 1\}$ representing the quasi-static/static state. The prior map is denoted as $\mathcal{M}_{t-1}$ and the current environment map observation as $\hat{\mathcal{M}}_t$. The aim for our system is to classify quasi-static features based on semantic labels and temporal persistence. As a result of classification, a static map denoted by $\hat{\mathcal{M}}_t^S$ is formed. The final output of the system is the refined lifelong map denoted as $\mathcal{M}_t$.

IV. SYSTEM OVERVIEW

The complete overview of the system is provided in Figure 2. The first step in our system involves aligning the current mapping session with the prior map $\mathcal{M}_{t-1}$ to generate the current map observation $\hat{\mathcal{M}}_t$. We assume that both maps contain point-wise semantic labels. In case the prior map $\mathcal{M}_{t-1}$ doesn't contain semantic labels, we perform nearest neighbor search to transfer semantic labels from the current map observation $\hat{\mathcal{M}}_t$ to the prior map $\mathcal{M}_{t-1}$. Next, we define a classification model to identify points from truly static objects. The classification model is used to build a factor graph operating on the maps $\mathcal{M}_{t-1}$ and $\hat{\mathcal{M}}_t$. The inference on this factor graph results in the classification of the objects' state in the maps, allowing us to build static maps $\mathcal{M}_{t-1}^S$ and $\hat{\mathcal{M}}_t^S$ for prior and current map observation respectively. The second part of our system deals with the refinement of 3D maps based on the changes detected from semantic voxel grids generated from the two static maps $\mathcal{M}_{t-1}^S$ and $\hat{\mathcal{M}}_t^S$. We classify the changes in each voxel into different categories to perform the update of the lifelong map.

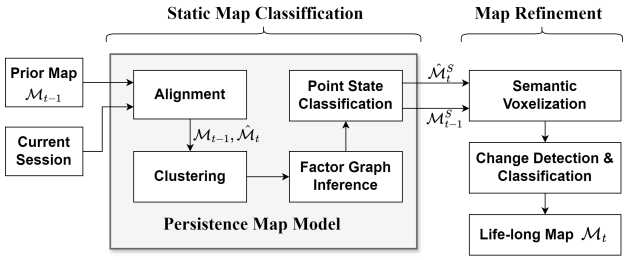


Fig. 2: Our system takes the prior map $\mathcal{M}_{t-1}$ and current mapping session as input. We aligns them to create the current map observation $\hat{\mathcal{M}}_t$ and perform persistence modeling for the two maps based on factor graphs. This results in static versions of two maps $\mathcal{M}_{t-1}^S$ and $\hat{\mathcal{M}}_t^S$ for map refinement to generate the updated lifelong map $\mathcal{M}_t$.

V. STATIC MAP CLASSIFICATION

The aim for the classification of 3D map objects is to distinguish between static, quasi-static and dynamic objects based on the temporal persistence and semantic information. While dynamic objects can be detected based on the motion across frames, quasi-static objects which exhibit temporary static state, require analysis over extended time involving temporal persistence over multiple maps. In this work, the classification is performed based on the persistency modeling for points within the map.

A. Persistence Model

For each feature $\mathbf{f}_i$ and time instance t, we define a binary random variable $X_t \in \{0, 1\}$ to indicate whether $\mathbf{f}_i$ *persists* at time t. We define an observation of $\mathbf{f}_i$ at some time $t_i \in [0, \infty)$ as the random variable $Z_{t_i} \in \{0, 1\}$, where $Z_{t_i} = 1$ indicates that $\mathbf{f}_i$ is observed at time t_i and vice versa. We define the set of observations taken up to time t as $\mathcal{Z}_t \triangleq \mathcal{Z}_{1:L} \triangleq \{Z_{t_i}\}_{i=1}^L$, where $t_i < t_{i+1}, \ldots, t_L \leq t$. Following the Bayesian survival analysis approach [36], we formulate *the survival time* $T \in [0, \infty)$ of $\mathbf{f}_i$ as a random variable, which represents the time duration from $\mathbf{f}_i$'s first appearance to its disappearance. Hence, we employ the persistence model from [31] as follows:

$$\begin{aligned} T &\sim f_T(\cdot), \\ X_t|T &= \begin{cases} 1, & t \leq T \\ 0, & t > T \end{cases}, \\ Z_t|X_t &\sim f_{Z_t}(\cdot|X_t), \end{aligned} \tag{1}$$

where $f_T(\cdot)$ and $f_{Z_t}(\cdot|X_t)$ are some user-defined probability distribution functions. To define the prior, an exponential distribution is applied:

$$f_T(t) = \lambda e^{-\lambda t}, \tag{2}$$

where λ is a user-defined "death-per-second" constant. The distribution $f_{Z_t}(\cdot|X_t)$ can be characterized using the mis-detection P_M and false-detection P_F probabilities as follows:

$$P_M = \mathcal{P}(Z_t = 0|X_t = 1), \;\; P_F = \mathcal{P}(Z_t = 1|X_t = 0). \tag{3}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Note that from (3) we also have complement probabilities

$$\bar{P}_M \triangleq \mathcal{P}(Z_t = 1|X_t = 1) = 1 - P_M,$$
$$\bar{P}_F \triangleq \mathcal{P}(Z_t = 0|X_t = 0) = 1 - P_F. \tag{4}$$

Our objective is to construct the probability distribution $\mathcal{P}(X_t = x|\mathcal{Z}_t)$ as a function of $x \in \{0, 1\}$, which will be used in the following optimization problem:

$$X_t^{\text{MPE}} = \arg\max_{x\in\{0,1\}} \mathcal{P}(X_t = x|\mathcal{Z}_{1:L}), \tag{5}$$

From Bayes' rule, we can obtain the following:

$$\mathcal{P}(X_t = x|\mathcal{Z}_{1:L}) = \frac{\mathcal{P}(\mathcal{Z}_{1:L}|X_t = x)\mathcal{P}(X_t = x)}{\mathcal{P}(\mathcal{Z}_{1:L})}, \tag{6}$$

Substituting (6) to (5), we can see that

$$\begin{aligned} X_t^{\text{MPE}} &= \arg\max_{x\in\{0,1\}} \frac{\mathcal{P}(\mathcal{Z}_{1:L}|X_t = x)\mathcal{P}(X_t = x)}{\mathcal{P}(\mathcal{Z}_{1:L})} \\ &= \arg\max_{x\in\{0,1\}} \mathcal{P}(\mathcal{Z}_{1:L}|X_t = x)\mathcal{P}(X_t = x), \end{aligned} \tag{7}$$

where the last equality is because $\mathcal{P}(\mathcal{Z}_{1:L})$ is a constant that does not affect the minima of the cost function at X_t^{MPE} (maximum probable explanation). Hence, we are only concerned with finding the probability distribution function $\mathcal{P}(\mathcal{Z}_{1:L}|X_t = x)$ and probability $\mathcal{P}(X_t = x)$. For the former, we parametrize the likelihood function $\mathcal{P}(\mathcal{Z}_{1:L}|X_t = x)$ to get the expression as:

$$\mathcal{P}(\mathcal{Z}_{1:L}|X_t = x) = \left[\prod_{i=0}^{L} P_M^{1-z_{t_i}} \bar{P}_M^{z_{t_i}}\right]^x \left[\prod_{i=0}^{L} P_F^{z_{t_i}} \bar{P}_F^{1-z_{t_i}}\right]^{1-x} \tag{8}$$

where, we have assumed that observations $\mathcal{Z}_{1:L}$ are mutually independent given the state of $X_t = x$. For probability $\mathcal{P}(X_t = x)$, the distribution $\mathcal{P}(T \leq t)$ where the event $X_t = 0$ is equivalent to the event $T \leq t$. The probability $\mathcal{P}(T \leq t)$ can be easily found by integrating $f_T(\cdot)$ in (1) as:

$$\mathcal{P}(T \leq t) = \int_0^t f_T(x)dx, \tag{9}$$

$$\mathcal{P}(T > t) = 1 - \int_0^t f_T(x)dx. \tag{10}$$

As a result, given M features we have M variables $\boldsymbol{X} = \{X_t^m | m = 1 \dots M\}$ and M sets of observations $\boldsymbol{Z} = \{\mathcal{Z}_t^m | m = 1 \dots M\}$ modeled by the Bayes network.

$$\mathcal{P}(\boldsymbol{X}|\boldsymbol{Z}) = \prod_{m=1}^{M} \mathcal{P}(X_t^m|\mathcal{Z}_{1:L_m}^m) \tag{11}$$

Remark 1. *Note that we view $\mathcal{P}(X_t|\mathcal{Z}_t)$ as a function over the variable X, with t, $\mathcal{Z}_t$, λ, P_M, P_F. $\mathcal{P}(X_t|\mathcal{Z}_t)$ as the function's parameters. We also will assume all $\mathcal{Z}^m$ sets will have the same size, i.e. $L_m = L$ for all m.*

B. Factor Graph for Persistence Estimation

For the persistence estimation problem, a robust and scalable framework for inference is provided through a factor graph $\mathcal{F} = (\mathcal{X}, \mathcal{V}, \mathcal{E})$ for map $\hat{\mathcal{M}}_t$ consisting of variable nodes $\mathcal{X}$, factors $\mathcal{V}$ and edges $\mathcal{E}$. Based on these, the factor graph defines the factorization of global function $\psi(\mathcal{X})$ where the joint distribution is expressed as product of factors. To derive the factor graph from Bayes net in (11), a variable X_i is dependent on two functions, i.e. likelihood $\mathcal{P}(\mathcal{Z}_{1:L}|X_t = x)$ and prior $\mathcal{P}(X_t = x)$. Then, the global function is expressed as the product of these individual functions for each variable.

$$\psi(\mathcal{X}) = \prod_{k}^{K} \mathcal{P}(\mathcal{Z}_{1:L}^k|X_t^k)\mathcal{P}(X_t^k) \tag{12}$$

where K is the number of variables. For the factor graph, the likelihood and prior distribution functions of a variable X_i are termed as factors $\psi(X_i)$ connected to it. The global function $\psi(\mathcal{X})$ represents an unnormalized distribution $\psi(\mathcal{X}) \propto \mathcal{P}(\mathcal{X})$. We derive the joint posterior probability for the given variables in the map modeled by the factor graph [37].

$$\mathcal{P}(\mathcal{X}) = \frac{1}{Q} \prod_{k}^{K} \psi_k(X_k), \tag{13}$$

where Q is the partition function for normalization. For each variable, we collapse the observations into a single measurement factor, resulting in the number of factors equal to the number of variables.

We perform inference for optimization of the factor graph [37], [38] and estimate the maximum probable explanation (MPE) using max product algorithm [37] for the variable within the factor graph.

$$\mathcal{X}^* = \arg\max_{\mathcal{X}} \mathcal{P}(\mathcal{X}) \tag{14}$$

The Bayes net in (11) and the derived factor graph in (13) are only able to classify points belonging to moving objects. Since, the sequences of observations used for the likelihood function $\mathcal{P}(\mathcal{Z}_{1:L}|X_t)$ belong to a single mapping session, it is unable to correctly detect temporary static objects. To accurately classify quasi-static objects, we extend the variable space by including observations from multiple maps. Given a prior map $\mathcal{M}_{t-1}$ and current map observation $\hat{\mathcal{M}}_t$, we apply the same persistence model from (1) and derive the factor graph from the Bayes net in (11). The variables $\mathcal{X} = \{\mathcal{X}_{t-1}, \mathcal{X}_t\}$ includes $\mathcal{X}_{t-1}$ representing the state of points in $\mathcal{M}_{t-1}$, and $\mathcal{X}_t$ for the variables in $\hat{\mathcal{M}}_t$. The joint probability of the factor can then be estimated as:

$$\mathcal{P}(\mathcal{X}) = \frac{1}{Q} \prod_{j}^{J} \psi_j^{t-1}(\mathcal{X}_{t-1}) \prod_{k}^{K} \psi_k^t(\mathcal{X}_t) \prod_{l}^{L} \eta_l(\mathcal{X}'_{t-1}, \mathcal{X}'_t), \tag{15}$$

$$\eta_l(\mathcal{X}'_{t-1}, \mathcal{X}'_t) = \begin{cases} \beta, & \mathcal{X}'_{t-1} = \mathcal{X}'_t \\ \gamma, & \mathcal{X}'_{t-1} \neq \mathcal{X}'_t \end{cases}, \tag{16}$$

where $\psi^{t-1}(X_{t-1})$ and $\psi^t(X_t)$ are the factors for the variables $\mathcal{X}^{t-1}$ and $\mathcal{X}^t$ respectively. Furthermore, we also introduce the factor $\eta_l(X'_{t-1}, X'_t)$ with pre-defined hyper-parameters $\beta = 0.8$ and $\gamma = 0.2$, for temporal consistency of the points that can be associated between the two maps.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

C. Persistence Modeling for large-scale Maps

The point-wise models for estimating the temporal persistence in the case of multi-map problem, are not suitable for large-scale maps as evident from earlier work [14]. Therefore to make our system robust and scalable to large-scale maps, we propose to adopt the persistence modeling to clusters of points instead of individual points within each map.

1) Clustering: The clustering process begins with selecting the points associated with semantic classes which portray quasi-static behavior such as cars, pedestrians, bikes, etc. Additionally points with high semantic label uncertainty reported from the semantic segmentation model are also included. These points are then clustered using DBSCAN [39]. It yields a set of clusters C_{t-1} for prior map $\mathcal{M}_{t-1}$ and C_t for current map observation $\hat{\mathcal{M}}_t$. Cluster correspondences are established based on Euclidean distance between their respective centroids. This results in a subset of clusters $C' = \{C'_{t-1}, C'_t\}$ which have association across the two maps.

2) Factor Graph based on Clustering: We assume that the clustering is correct and apply the persistence model from Equation (1) to the cluster variables. We represent the state of each cluster c_i by $S_i \in \{0, 1\}$ representing the quasi-static/static state for the points within that cluster. We can estimate the posterior probability $\mathcal{P}(S_t|\mathcal{Z})$ from Equation (6). Subsequently, we build the factor graph based on cluster variables across two maps. Following from Equation (15), the posterior probability for the cluster-based factor graph is given as follows:

$$\mathcal{P}(\mathcal{S}) = \frac{1}{Q}\prod_{j}^{J}\psi_j^{t-1}(\mathcal{S}_{t-1})\prod_{k}^{K}\psi_k^{t}(\mathcal{S}_t)\prod_{l}^{L}\eta_l(\mathcal{S}'_{t-1}, \mathcal{S}'_t) \tag{17}$$

where, $\mathcal{S} = \{\mathcal{S}_{t-1}, \mathcal{S}_t\}$, $\mathcal{S}_{t-1}$ for prior map clusters and $\mathcal{S}_t$ for current map observation clusters. We perform inference to find the maximum probable explanation for the optimized state $\mathcal{S}^*$ of the clusters. The points within each cluster are then classified as static or quasi-static based on the values of $\mathcal{S}^*_{t-1}$ for current clusters and $\mathcal{S}^*_t$ for prior clusters. We aggregate all the static points to generate prior static map $\mathcal{M}^S_{t-1}$ and current static map $\hat{\mathcal{M}}^S_t$.

VI. Map Refinement

A. Semantic Voxelization

For our system, we perform voxelization to effectively analyze changes and refine the 3D maps $\mathcal{M}^S_{t-1}$ and $\hat{\mathcal{M}}^S_t$. This discretization process preserves the geometric and semantic information while providing an efficient way of comparing large-scale maps. The voxelization results in voxel grids $V_{t-1} = \{v_1^{t-1}, v_2^{t-1}, ..., v_u^{t-1}\}$ and $V_t = \{v_1^t, v_2^t, ..., v_v^t\}$ for prior $\mathcal{M}^S_{t-1}$ and current $\hat{\mathcal{M}}^S_t$ static maps respectively. For each voxel v_i, we determine the semantic label based on voting of its constituent points. Additionally, we compute the centroid $\mathbf{m}_{v_i}$ and average normal $\mathcal{N}_{v_i}$ for each voxel. The semantic labels, centroid $\mathbf{m}_{v_i}$ and normal $\mathcal{N}_{v_i}$ are vital for the change detection between two grids.

B. Voxel Grid Comparison

To perform the change detection between the two voxel grids V_{t-1} and V_t, we rely on the overlapping region between the two grids denoted as V^o_{t-1} and V^o_t. The change detection criteria is defined based on the geometric and semantic similarity between the voxels of each grid. As a first step, we define the distance threshold τ_d and normal threshold $\tau_{\mathcal{N}}$. We analyze each voxel from V^o_t by searching for its nearest voxel in V^o_{t-1} using KD-tree. If the Euclidean distance between the centroids of two voxels exceeds the distance threshold τ_d, it is considered as a change candidates. Further check is performed based on normals and semantic labels. The voxels are confirmed as changed if the cosine similarity of the normals of two voxels falls below the normal threshold $\tau_{\mathcal{N}}$ or their semantic labels are different. Similarly, we analyze the voxels from the V^o_{t-1} against the voxels in V^o_t. During this comparison the change status of each voxel is recorded in boolean arrays A_{t-1} and A_t for the grids V^o_{t-1} and V^o_t respectively. The two voxel grids along with their respective change statuses help to perform accurate classification.

C. Change Classification

After the voxel grids comparison, we get arrays A_{t-1} and A_t for the voxels V^o_{t-1} and V^o_t. Each index in the array A corresponds to a voxel from the the grid V^o, where a value of $true$ represents change in that voxel. We perform a classification of the voxels into five categories as follows:

1) Previously Explored Voxels V_{pe}: These are the voxels that are present in V_{t-1} but not in V_t. Therefore, we define the previously explored set of voxels V_{pe}.

$$V_{pe} = V_{t-1}/V_t = \{v \in V_{t-1} | v \notin V_t\} \tag{18}$$

2) Newly Explored Voxels V_{ne}: Newly explored voxels V_{ne} are present in the grid V_t but not seen in the prior static map grid V_{t-1}.

$$V_{ne} = V_t/V_{t-1} = \{v \in V_t | v \notin V_{t-1}\} \tag{19}$$

3) Co-existing Voxels V_c: The coexisting voxels correspond to the voxels from V^o_{t-1} and V^o_t, that are not changed. Using the change detection results A_{t-1} and A_t, we define the co-existing voxels as follows:

$$V_c = \{v_i \in \{V^o_{t-1}, V^o_t\} | \neg A_{t-1}(i) \wedge \neg A_t(i)\} \tag{20}$$

4) Obsolete Voxels V_{ob}: The obsolete voxels from V^o_{t-1} are the voxels marked true in the change index array A_{t-1}.

$$V_{ob} = \{v_i \in V^o_{t-1} | A_{t-1}(i) = True\} \tag{21}$$

5) Emerging Voxels V_e: Similarly, the emerging voxels are given as the voxels from V_t with $True$ label from A_t.

$$V_e = \{v_i \in V^o_t | A_t(i) = True\} \tag{22}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE I: Specifications of the experiments used for evaluation in this work. For NTU dataset experiments E_5 and E_6, we apply map-maintenance on the lifelong map produced from experiments E_3 and E_4 respectively.

Campus	Experiment	Prior Map			Current Observation		
		Map Data	Collection Date	LiDAR Type	Sequence	Collection Date	LiDAR Type
KTH	E_1	Survey-grade	Sep-2022	Leica-RTC360	mcdll-kth	Nov-2024	OS1-64 line
	E_2	MCD kth_day_06 & kth_day_10	Nov-2022	OS1-64 line	mcdll-kth	Nov-2024	OS1-64 line
NTU	E_3	Survey-grade	Jan-2022	Leica-RTC360	mcdll-ntu01	Sep-2024	Livox mid-360
	E_4	MCD ntu_day_01	Feb-2022	OS1-128 line	mcdll-ntu01	Sep-2024	Livox mid-360
	E_5	E3 LL Map	Sep-2024	-	mcdll-ntu02	Dec-2024	Livox mid-360
	E_6	E4 LL Map	Sep-2024	-	mcdll-ntu02	Dec-2024	Livox mid-360

D. Map Update

The final output of the system is an updated lifelong map $\mathcal{M}_t$ is constructed by aggregating points belonging to voxels of categories V_{pe}, V_{ne}, V_c and V_e. The points corresponding to the obsolete voxels V_{ob} are saved separately to preserve the changes in the environment as a delta map $\mathcal{M}_\Delta$. The delta map $\mathcal{M}_\Delta$ is kept for each iteration of lifelong mapping as "log" of the changes.

VII. Experiments Details

The complete details of the experiments performed in this works are given in Table I. We generate prior maps from the MCD dataset [16] in the experiments for this paper. MCD dataset was collected in three locations including Hamburg, KTH main campus Stockholm and NTU campus Singapore. For this work, we use KTH and NTU sequences. The dataset includes a survey grade ground truth map along with LiDAR data for each campus. The LiDAR data was collected with Ouster 128-line for NTU and 64-line for KTH campus.

For lifelong experiments, we collect one sequence from KTH campus with Ouster OS1 64-line LiDAR and and two sequences on NTU from [3] with Livox mid-360, captured over a four-month gap. We refer to the new dataset as MCDll dataset (Multi-campus lifelong dataset) comprising of three sequences mcdll-kth, mcdll-ntu01 and mcdll-ntu02 sequences. We collect NTU dataset in two sequences to illustrate the lifelong mapping process in multiple iterations and in the process capturing the construction of large building at NTU campus. The detailed specifications of the dataset are given in Table I. There is a time gap of more than two years between the MCD dataset and MCDll dataset. We observed several changes in the new dataset sequences, including large changes such as a new building at the NTU campus, main entre building of KTH campus covered by wooden boards for renovation and many others. We perform two lifelong mapping experiments E_1 and E_2 for KTH campus using survey-grade and SLAM map as prior maps respectively. For NTU campus, we perform four experiments E_3 to E_6. We perform map maintenance on survey-grade prior map using mcdll-ntu01 sequence in experiment E_3 as first iteration. In the second iteration, we use the lifelong map from E_3 and apply maintenance with mcdll-ntu02 sequence in E_5. Similarly, we perform map maintenance in experiments E_4 and E_6 with SLAM map as prior map. We benchmark the performance of our system against LT-mapper [12] and ELite [14]. For the experiments E_1, E_3, and E_5 we use the survey-grade LiDAR scans and poses given in MCD dataset [16].

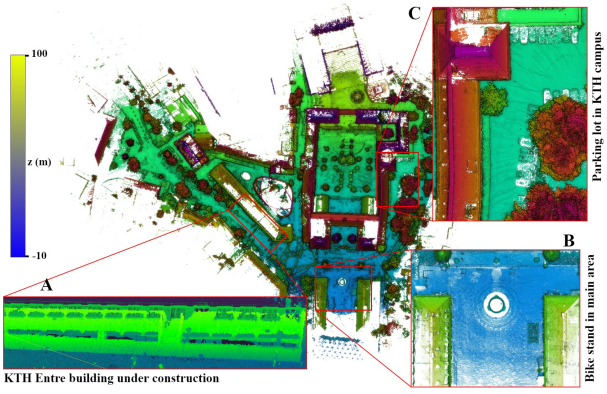


Fig. 3: Lifelong map for KTH campus produced using our method in experiment E_1 with survey map given as prior map. A consistent lifelong map is produced owing to the detail-oriented approach of our method.

VIII. Evaluation

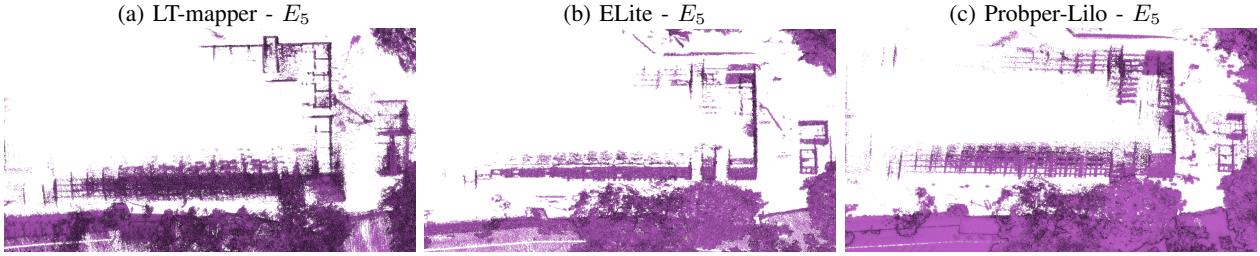
A. Qualitative Evaluation

The lifelong map for E_1 using our method is shown in fig. 3. We highlight three areas in the KTH campus with large changes. The Entre building was undergoing renovation and it was covered by wooden boards (A), a new bike stand was constructed in the main area (B) and parking lot where frequent quasi-static objects are observed (C). The lifelong map produced by our method accurately demonstrates these changes. The Entre building is correctly updated to show the construction wall in front of the building with the original structure also preserved. Next, the bike stand is also updated accurately and all quasi-static objects were correctly removed in the parking area. Due to the page limit, the results of experiments E_1 and E_2 are discussed in details in the supplementary material [40].

The evaluation of the NTU dataset is performed in four experiments E_3, E_4, E_5 and E_6. For the NTU campus, major changes were observed with the construction of a new building. In mcdll-ntu01, the building is only half-constructed, and sequence mcdll-ntu02 was collected a few months later when the building was complete. For experiments E_3 and E_4, the mcdll-ntu01 sequence is used to update the respective prior maps. Then the resulting lifelong maps are updated using mcdll-ntu02 sequence. The building area for the final lifelong maps generated in experiments E_5 and E_6 using the three methods is shown in Figure 4. LT-mapper is able to update the changed part, but results in a distorted lifelong map in both experiments. When a prior map was used for

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

(a) LT-mapper - E_5 (b) ELite - E_5 (c) Probper-Lilo - E_5



(d) LT-mapper - E_6 (e) ELite - E_6 (f) Probper-Lilo - E_6

Fig. 4: The results of experiments E_5 and E_6. The lifelong map for LT-mapper [12] shows the building but it is distorted, and for the lifelong map generated from ELite [14] some parts of the building are missing. Our method generated the lifelong map with accurate and consistent representation of the building for both experiments.

ELite, it was unable to detect the changes as the majority of the prior map was returned unchanged in lifelong map. In Figure 4b, we can see the structure from from the prior map and the new building point are overlayed on top of it. For experiment E_6, ELite correctly updated the building are but most of the building point are removed labeled as dynamic points. Lifelong maps generated by Probper-Lilo are shown in Figure 4c and Figure 4f. Our method is able to generate an accurate and consistent representation for both experiment with survey grade map and SLAM maps.

B. Quantitative Evaluation

We apply the map quality evaluation method proposed in [41], using accuracy (AC), completeness (COM), Chamfer distance (CD), mean map entropy (MME), and average Wasserstein distance (AWD). The vital part of a lifelong mapping algorithm to maintain global and local consistency. The map evaluation metrics are given in Table II for the six mapping experiments. LT-mapper shows high accuracy on E_2, E_4, and E_6 experiments with low completeness score as the accuracy is only accounted for the inlier points. In such cases, a better metrics is the AWD where it report high values. ELite gives very good consistency for the experiments E1, E3 and E5, because it returns most of the prior map unchanged, as we saw in qualitative evaluation that when survey grade prior map is used, ELite was unable to detect and update the changes. For the other experiments, ELite gave better consistency as compared to LT-mapper but much lower than our method. One issue observed with ELite is that during dynamic object removal a significant part of static objects were also removed, which affects the resulting lifelong map's consistency. Finally, our method gave excellent accuracy with high COM scores. CD and AWD highlight local and global consistency for lifelong maps using our method. Our method reported higher consistency for all experiments with SLAM and survey grade maps, showcasing the effectiveness and strong generalization of our system. We also provide an evaluation of the dynamic object detection of our system in the supplementary material [40].

TABLE II: Map quality evaluation for the final maps produced for LT-mapper, ELite and our method for the MCD and MCDll dataset sequences.

		KTH			NTU		
		E_1	E_2	E_3	E_4	E_5	E_6
AC ↓	lt-mapper	0.048	**0.063**	0.042	**0.066**	0.043	**0.060**
	ELite	0.044	0.211	0.040	0.220	0.051	0.216
	ProbPer-Lilo	**0.025**	0.075	**0.014**	0.086	**0.017**	0.086
COM ↑	lt-mapper	80.92	23.43	81.40	18.03	84.04	5.47
	ELite	88.50	46.43	91.05	20.98	82.87	33.36
	ProbPer-Lilo	**95.08**	**83.62**	**94.33**	**69.73**	**96.27**	**77.40**
CD ↓	lt-mapper	0.578	1.683	0.717	4.55	0.671	12.77
	ELite	**0.234**	2.581	0.198	10.94	0.286	5.461
	ProbPer-Lilo	0.345	**0.393**	**0.140**	**1.098**	**0.116**	**0.960**
MME ↓	lt-mapper	-8.79	-8.21	-8.82	-8.31	-8.72	-8.20
	ELite	-7.82	-5.79	-8.16	-5.79	-7.76	-5.73
	ProbPer-Lilo	**-8.81**	**-8.23**	**-10.11**	**-8.27**	**-9.82**	**-8.26**
AWD ↓	lt-mapper	0.326	0.828	0.917	0.945	0.281	1.112
	ELite	0.231	0.789	0.247	0.909	0.306	1.045
	ProbPer-Lilo	**0.089**	**0.373**	**0.128**	**0.592**	**0.101**	**0.485**

IX. CONCLUSIONS

Lifelong mapping enables the usage of point cloud map for several applications including robotics, autonomous vehicles and surveying. We have proposed a complete lifelong mapping framework that takes multiple sessions maps and perform static structure classification and then map refinement based on the static maps. The static map classification is based on quasi-static modeling. We proposed a novel discrete probabilistic method to model the state of points within a map leveraging a factor graph. Our system was evaluated on large-scale real-world dataset with significant changes in the environment, and demonstrated state-of-the-art performance.

REFERENCES

[1] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022. 1

[2] T.-M. Nguyen, D. Duberg, P. Jensfelt, S. Yuan, and L. Xie, "Slict: Multi-input multi-scale surfel-based lidar-inertial continuous-time odometry and mapping," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2102–2109, 2023. 1

[3] J. Li, S. Yuan, M. Cao, T.-M. Nguyen, K. Cao, and L. Xie, "Hcto: Optimality-aware lidar inertial odometry with hybrid continuous time optimization for compact wearable mapping system," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 211, pp. 228–243, 2024. 1, 6

[4] *Leica RTC360 3D Reality Capture Solution—Fast Agile Precise*, Leica, 4 2022. [Online]. Available: https://leica-geosystems.com/products/laser-canners/scanners/leica-rtc360 1

[5] K. Konolige and J. Bowman, "Towards lifelong visual maps," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 1156–1163. 1

[6] R. Elvira, J. D. Tardós, and J. M. Montiel, "Orbslam-atlas: a robust and accurate multi-map system," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6253–6259. 1

[7] A. Cramariuc, L. Bernreiter, F. Tschopp, M. Fehr, V. Reijgwart, J. Nieto, R. Siegwart, and C. Cadena, "maplab 2.0–a modular and multi-modal mapping framework," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 520–527, 2022. 1

[8] Y. Cui, R. Chen, W. Chu, L. Chen, D. Tian, Y. Li, and D. Cao, "Deep learning for image and point cloud fusion in autonomous driving: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 722–739, 2021. 1

[9] Y. Ren, F. Zhu, G. Lu, Y. Cai, L. Yin, F. Kong, J. Lin, N. Chen, and F. Zhang, "Safety-assured high-speed navigation for mavs," *Science Robotics*, vol. 10, no. 98, p. eado6187, 2025. 1

[10] F. Zhu, Y. Ren, L. Yin, F. Kong, Q. Liu, R. Xue, W. Liu, Y. Cai, G. Lu, H. Li, *et al.*, "Swarm-lio2: Decentralized, efficient lidar-inertial odometry for uav swarms," *IEEE Transactions on Robotics*, 2024. 1

[11] Y. Ren, Y. Cai, F. Zhu, S. Liang, and F. Zhang, "Rog-map: An efficient robocentric occupancy grid map for large-scene and high-resolution lidar-based motion planning," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 8119–8125. 1

[12] G. Kim and A. Kim, "Lt-mapper: A modular framework for lidar-based lifelong mapping," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7995–8002. 1, 2, 6, 7

[13] L. Yang, S. M. Prakhya, S. Zhu, and Z. Liu, "Lifelong 3d mapping framework for hand-held & robot-mounted lidar mapping systems," *IEEE Robotics and Automation Letters*, 2024. 1, 3

[14] H. Gil, D. Lee, G. Kim, and A. Kim, "Ephemerality meets lidar-based lifelong mapping," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Atlanta, May. 2025. 1, 3, 5, 6, 7

[15] B. C. Sanders, R. C. Nelson, and R. Sukthankar, "A theory of the quasi-static world," in *2002 International Conference on Pattern Recognition*, vol. 3. IEEE, 2002, pp. 1–6. 2

[16] T.-M. Nguyen, S. Yuan, T. H. Nguyen, P. Yin, H. Cao, L. Xie, M. Wozniak, P. Jensfelt, M. Thiel, J. Ziegenbein, and N. Blunder, "Mcd: Diverse large-scale multi-campus dataset for robot perception," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6 2024. [Online]. Available: https://mcdviral.github.io/ 2, 6

[17] D. Duberg, Q. Zhang, M. Jia, and P. Jensfelt, "DUFOMap: Efficient dynamic awareness mapping," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 1–8, 2024. 2

[18] G. Kim and A. Kim, "Remove, then revert: Static point cloud map construction using multiresolution range images," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 758–10 765. 2

[19] H. Lim, S. Hwang, and H. Myung, "Erasor: Egocentric ratio of pseudo occupancy-based dynamic object removal for static 3d point cloud map building," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2272–2279, 2021. 2

[20] H. Lim, L. Nunes, B. Mersch, X. Chen, J. Behley, H. Myung, and C. Stachniss, "Erasor2: Instance-aware robust 3d mapping of the static world in dynamic scenes." in *Robotics: Science and Systems*, 2023. 2

[21] M. Jia, Q. Zhang, B. Yang, J. Wu, M. Liu, and P. Jensfelt, "Beautymap: Binary-encoded adaptable ground matrix for dynamic points removal in global maps," *IEEE Robotics and Automation Letters*, 2024. 2

[22] 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*, vol. 7, no. 3, pp. 7503–7510, 2022. 2

[23] J. Sun, Y. Dai, X. Zhang, J. Xu, R. Ai, W. Gu, and X. Chen, "Efficient spatial-temporal information fusion for lidar-based 3d moving object segmentation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 11 456–11 463. 2

[24] H. Wei, R. Li, Y. Cai, C. Yuan, Y. Ren, Z. Zou, H. Wu, C. Zheng, S. Zhou, K. Xue, *et al.*, "Large-scale multi-session point-cloud map merging," *IEEE Robotics and Automation Letters*, 2024. 2

[25] X. Hu, J. Wu, J. Jiao, B. Jiang, W. Zhang, W. Wang, and P. Tan, "Ms-mapping: an uncertainty-aware large-scale multi-session lidar mapping system," *arXiv preprint arXiv:2408.03723*, 2024. 2

[26] U. Stilla and Y. Xu, "Change detection of urban objects using 3d point clouds: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 197, pp. 228–255, 2023. 2

[27] J. Gehrung, M. Hebel, M. Arens, and U. Stilla, "A fast voxel-based indicator for change detection using low resolution octrees," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, pp. 357–364, 2019. 2

[28] E. Langer, T. Patten, and M. Vincze, "Robust and efficient object change detection by combining global semantic information and local geometric verification," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 8453–8460. 2

[29] A. Adam, T. Sattler, K. Karantzalos, and T. Pajdla, "Objects can move: 3d change detection by geometric transformation consistency," in *European Conference on Computer Vision*. Springer, 2022, pp. 108–124. 2

[30] J. Rowell, L. Zhang, and M. Fallon, "Lista: Geometric object-based change detection in cluttered environments," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 3632–3638. 2

[31] D. M. Rosen, J. Mason, and J. J. Leonard, "Towards lifelong feature-based mapping in semi-static environments," in *2016 IEEE International conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1063–1070. 2, 3

[32] F. Nobre, C. Heckman, P. Ozog, R. W. Wolcott, and J. M. Walls, "Online probabilistic change detection in feature-based maps," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3661–3668. 2

[33] T. Deng, H. Xie, J. Wang, and W. Chen, "Long-term visual simultaneous localization and mapping: Using a bayesian persistence filter-based global map prediction," *IEEE Robotics & Automation Magazine*, vol. 30, no. 1, pp. 36–49, 2023. 2

[34] M. Saavedra-Ruiz, S. B. Nashed, C. Gauthier, and L. Paull, "Perpetua: Multi-hypothesis persistence modeling for semi-static environments," *arXiv preprint arXiv:2507.18808*, 2025. 2

[35] R. Ambruş, N. Bore, J. Folkesson, and P. Jensfelt, "Meta-rooms: Building and maintaining long term spatial models in a dynamic world," in *2014 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2014, pp. 1854–1861. 2

[36] J. G. Ibrahim, M.-H. Chen, and D. Sinha, *Bayesian survival analysis*. Springer Science & Business Media, 2001. 3

[37] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009. 4

[38] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on information theory*, vol. 47, no. 2, pp. 498–519, 2002. 4

[39] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, no. 34, 1996, pp. 226–231. 5

[40] Supplementary material. [Online]. Available: https://github.com/KTH-RPL/LifelongProbMapper/blob/main/Supplementary.pdf 6, 7

[41] X. Hu, J. Wu, M. Jia, H. Yan, Y. Jiang, B. Jiang, W. Zhang, W. He, and P. Tan, "Mapeval: Towards unified, robust and efficient slam map evaluation framework," vol. 10, no. 5, pp. 4228–4235, 2025. 7