Robotics: Science and Systems 2023
Daegu, Republic of Korea, July 10-July 14, 2023

POV-SLAM: Probabilistic Object-Aware Variational SLAM in Semi-Static Environments

Jingxing Qian[1], Veronica Chatrath[1,2], James Servos[3], Aaron Mavrinac[3], Wolfram Burgard[4], Steven L. Waslander[1], Angela P. Schoellig[1,2]

***Abstract*—Simultaneous localization and mapping (SLAM) in slowly varying scenes is important for long-term robot task completion. Failing to detect scene changes may lead to inaccurate maps and, ultimately, lost robots. Classical SLAM algorithms assume static scenes, and recent works take dynamics into account, but require scene changes to be observed in consecutive frames. Semi-static scenes, wherein objects appear, disappear, or move slowly over time, are often overlooked, yet are critical for long-term operation. We propose an object-aware, factor-graph SLAM framework that tracks and reconstructs semi-static object-level changes. Our novel variational expectation-maximization strategy is used to optimize factor graphs involving a Gaussian-Uniform bimodal measurement likelihood for potentially-changing objects. We evaluate our approach alongside the state-of-the-art SLAM solutions in simulation and on our novel real-world SLAM dataset captured in a warehouse over four months. Our method improves the robustness of localization in the presence of semi-static changes, providing object-level reasoning about the scene.**

I. Introduction

Simultaneous Localization and Mapping (SLAM) estimates a robot's pose within its environment, while at the same time creating a map of its surroundings. SLAM allows for autonomous navigation in GPS-denied situations, such as underground mines, office spaces, and warehouses. Many such tasks require robots to reliably repeat their trajectories over an extended period. However, most existing SLAM methods adopt the static world assumption [1, 2, 3, 4] which typically does not hold in the real world, as scenes are subject to change from human or robot activity. For example, a scene may contain dynamic objects (e.g., forklift driving within a factory) and semi-static objects that change position over time (e.g., pallets, boxes). Lacking the ability to properly handle such changes might result in catastrophic failures such as corrupted maps, divergent pose estimations, and obstacle collisions. Such potential failures emphasize the importance of robust SLAM solutions in the presence of scene dynamics in order to achieve efficient and robust long-term robotic operation.

[1]The University of Toronto Institute for Aerospace Studies and the University of Toronto Robotics Institute.
Emails: {firstname.lastname}@robotics.utias.utoronto.ca
[2]The Technical University of Munich.
Emails: {firstname.lastname}@tum.de
[3]Clearpath Robotics, Waterloo, Canada.
Emails: {jservos,amavrinac}@clearpath.ai
[4]The Technical University of Nuremberg.
Email: wolfram.burgard@utn.de

This work was supported by the Vector Institute for Artificial Intelligence in Toronto and the NSERC Canadian Robotics Network (NCRN).

Dataset download and Supplementary Material are available at https://github.com/Viky397/TorWICDataset

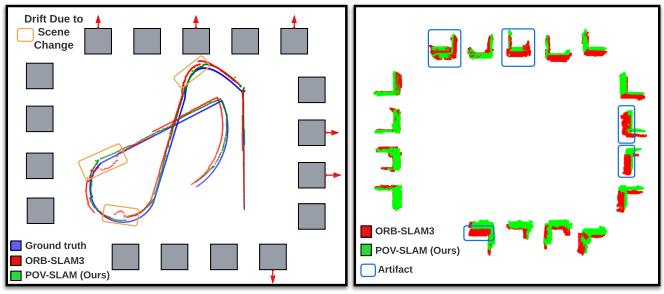

Fig. 1: A qualitative comparison of the trajectory and scene reconstruction of a semi-static synthetic scene, BoxSim, where the boxes shift when not in the camera's field of view (indicated by the red arrows). Here we compare POV-SLAM to a state-of-the-art SLAM solution, ORB-SLAM3 [5]. As ORB-SLAM3 is a feature-based method, we feed its pose estimates into a semi-static mapping method, POCD [6], to generate the map. ORB-SLAM3 assumes a static world, suffering from localization drift (orange boxes, left) when changed objects are encountered, which leads to artifacts and incorrect map updates (blue boxes, right). Our approach explicitly infers object-level scene changes and provides more robust long-term localization and dense reconstruction. The reference desired reconstruction is shown in the top row of Figure 8.

Recent works have attempted to handle dynamic environments in one of two ways. The first strategy leverages semantic and geometric information to mask out all potentially dynamic objects, treating them as outliers [7, 8, 9, 10, 11, 12]. Hence, the system only tracks against the static background, though it often covers a small portion of the sensor's field-of-view (FOV) in cluttered environments. The second strategy builds a model for each detected object. The system then either tracks the camera against the static background and refines the object models in a two-step pipeline, or performs camera and object tracking in a joint optimization problem [13, 14, 15, 16]. However, the second strategy requires motion to be detected over consecutive frames, and long-term, semi-static changes where objects shift, disappear, or appear in the scene, have not been thoroughly studied in SLAM. Recent attempts to handle semi-static changes during map maintenance extend object-centric mapping methods to explicitly consider semi-static changes by estimating a consistency score for each object from a known robot pose [6, 17, 18, 19]. Critically, when the robot pose is unknown, object consistency is difficult to calculate. The aforementioned consistency estimation methods can lead to multiple ambiguous and sub-optimal solutions. This limitation highlights the need for a statistically consistent method to infer both the robot pose and object consistency.

We tackle the challenge of simultaneous localization and object-level change detection in large semi-static scenes. We follow an object-aware strategy, as most mobile robots operate in environments consist of rigid objects that move continuously or change location between visits. In addition to pose estimation, an up-to-date, object-level dense reconstruction is desired to provide rich geometric information for downstream tasks (e.g., perception-aware planning and control). We introduce a novel framework, POV-SLAM, which leverages recent works on object-level Bayesian consistency estimation for semi-static scenes [6], to tackle the challenge in a joint optimization problem. We derive a variational formulation to approximate the Gaussian-Uniform measurement model of potentially-changing objects, and use expectation-maximization (EM) to guarantee improvement of the evidence lower bound (ELBO) of a factor graph SLAM problem. At every EM iteration, the object consistencies and the robot poses are refined using geometric and semantic measurements.

Additionally, there is a lack of SLAM datasets for long-term localization and mapping in large, semi-static environments. In collaboration with Clearpath Robotics, we present a real-world semi-static SLAM dataset in a warehouse with dynamic and semi-static changes that occur over four months. To facilitate easier performance evaluation, we provide high-quality 3D scans of the entire warehouse and the ground truth robot trajectories, obtained from a Leica MultiStation and an onboard Ouster 128-beam LiDAR.

Our proposed method is evaluated on: a 2D simulation to demonstrate the probabilistic framework in action and justify design choices, a synthetic semi-static dataset, and our real-world warehouse dataset. We analyze the reconstruction quality relative to a state-of-the-art (SOTA) dense semi-static mapping method [6] and compare the localization accuracy against a SOTA feature-based SLAM method [5] as well as a semi-static object-level SLAM [17] approach. We show that our framework is robust to semi-static changes in the scene. The main contributions of our paper are:

- We derive a variational formulation for the Gaussian-Uniform bimodal measurement likelihood of potentially-changing objects. It exploits the Bayesian object consistency update rule introduced in [6] and provides an evidence lower bound (ELBO) for efficient inference.
- We introduce an expectation-maximization (EM) algorithm to optimize factor graphs involving the variational measurement model for potentially-changing objects.
- We design POV-SLAM, an object-aware, factor graph SLAM pipeline that tracks and reconstructs semi-static object-level changes. POV-SLAM builds on top of the SOTA SLAM [5] and semi-static mapping [6] methods, and uses our variational EM (VEM) strategy. The system is demonstrated both in simulation and in the real world.
- We release a new SLAM dataset captured in a warehouse over four months. The environment contains static, semi-static, and dynamic objects as seen by RGB-D cameras and a 3D LiDAR. We also release a high-quality 3D scan of the warehouse and ground truth robot trajectories.

In Section II, we review the SLAM methods for changing scenes. In Section III, we present the key modules of the POV-SLAM pipeline. In Section IV, we derive the variational measurement model and discuss the details of our VEM algorithm. Finally, we evaluate POV-SLAM in both simulated and real-world experiments in Section V. To the best of our knowledge, our method is the first to achieve joint localization and object-level change detection for large, semi-static environments.

II. Related Works

A. Visual SLAM

Visual SLAM is a well-established type of SLAM, mainly achieved via either feature-based methods [1, 20, 21, 22] or dense methods [23, 4]. Sparse methods match feature points of images, having lighter computational requirements, focusing on localization, whereas dense methods seek to construct accurate and more complete representations of the environment, useful for navigation and collision avoidance.

In recent years, feature-based SLAM methods have gained traction for use with mobile robots in large environments, as they exhibit a high level of accuracy and efficiency. The seminal works of Mur-Artal *et al.* in ORB-SLAM [1] introduce a monocular, feature-based SLAM system with real-time camera relocalization. ORB-SLAM2 [24] and ORB-SLAM3 [5] extend [1] with stereo and RGB-D information. ORB-SLAM remains a state-of-the-art feature-based method [25, 26] and is extended to aid with our localization and map update strategy.

However, most current visual SLAM methods focus on static scenes, simply rejecting inconsistent landmarks from dynamic objects as outliers. As well, object-level scene information is ignored, resulting in inconsistent map updates when items move between robot passes. Our framework aims to use object-level understanding to track scene changes and aid with accurate localization in evolving scenes.

B. Dynamic SLAM

Dynamics and object-level reasoning in SLAM have been recently studied, and there exist two common strategies to handle changes. The first is to identify dynamics from input data, which can be extracted with a semantic segmentation network such as Mask R-CNN [27], discarding it completely [7, 8, 9, 10, 11, 12, 28]. Though this method is effective in the presence of a few dynamic objects, in cluttered environments, the static background is often only a small part of the sensor's FOV and ignoring all dynamic objects could lead to an insufficient number of visual features for localization.

The second strategy is to track the dynamic objects explicitly, which can be achieved using multi-object tracking (MOT) [13, 14, 15, 29, 30]. DetectFusion [14] uses semantic segmentation and motion consistency to extract both known and unknown objects. The work of Barsan *et al.* [29] uses instance-aware semantic segmentation and sparse scene flow to classify objects based on their activity. MID-Fusion [15] and EM-Fusion [30] obtain object masks and construct a signed distance function (SDF) model for objects from depth information. Object poses are obtained by directly aligning

depth measurements to their corresponding SDF models. VDO-SLAM [31] and ClusterSLAM [32] group landmarks to form objects and exploit rigid body motion to construct a factor graph, jointly solving for robot and object poses. The aforementioned methods require scene changes to be observed in consecutive frames, rendering this strategy ineffective under changes that occur over a long time horizon.

C. Semi-Static SLAM

SLAM in semi-static scenes is a difficult yet overlooked problem, that is crucial for long-term operation. One challenge in the presence of semi-static objects is ambiguity in the system state, caused by potential symmetry in the scene changes and the lack of continuously observed motion.

Recent works on map maintenance involving semi-static objects all aim to estimate a consistency score, based on given robot poses, to determine which part of the map needs to be updated. Fehr *et al.* [18] update an SDF map by calculating voxel-level differences between signed distance functions of the stored map and incoming depth measurements. Schmid *et al.* [19] maintain a set of object-level SDF sub-maps, propagating a stationarity score for each sub-map by calculating the overlap between their depth measurements and the existing map. Though intuitive, these overlap-based estimation methods are prone to localization errors. Gomez *et al.* [33] model objects as cuboid bounding volumes and construct an object factor graph to estimate the object poses and their moveability scores in offline batch optimization. To obtain a more accurate and consistent object-level consistency score at runtime, Qian *et al.* propose [6], a Bayesian update rule to iteratively propagate a probabilistic object state model using both geometric and semantic measurements, which was shown to be more robust against localization noise. However, the aforementioned incremental mapping solutions all assume reliable robot poses are given.

Walcott *et al.* propose a 2D LiDAR SLAM solution [34] that maintains a set of sub-maps for each region. The active sub-map is replaced with new measurements if there are inconsistencies and then stacked to form the final map. Rosen *et al.* incorporate a recursive Bayesian persistence filter [35] into classic feature-based SLAM systems to estimate the consistency of each point feature. In a more recent work, Ren *et al.* [17] attempt to integrate object-level consistency estimation and 3D visual SLAM in the presence of semi-static and dynamic objects. The authors first perform dense visual SLAM using the static background to estimate the camera pose. They calculate the image-plane overlap between new object measurements and their previously mapped objects, reconstructing the object if the inconsistency is large. The visual features of unobserved mapped objects and new object observations are compared to perform association and relocalization. However, a known static background is required to track the camera motion, making this method a two-step process and rendering it unstable in the presence of a large number of potentially changing objects.

Rogers *et al.* [36] and Xiang *et al.* [37] use an EM approach to handle semi-static point landmarks. The authors integrate the traditional landmark measurement model with a latent confidence score to weight its contribution in the cost function. The EM scheme is used to iteratively update the robot pose, landmark positions, and confidence scores. However, since the optimization process runs over the entire trajectory and rejection decisions are made based on a predefined threshold at the end of the process, these two methods are limited to offline settings. In our sliding window setup, landmark rejection decisions are revised probabilistically at every EM iteration during run-time, leading to more robust, fast, and accurate convergence. EM-based algorithms have been applied to other components of SLAM systems, such as data association [38].

III. System Description

A. Overview and Assumptions

This work focuses on long-term SLAM in the presence of semi-static objects. We aim to simultaneously localize the robot, and propagate a consistency estimate for each object. The robot localizes itself against objects with high measurement likelihoods, with changed objects being reconstructed once sufficient observations have been made. Finally, a truncated signed distance function (TSDF) map is produced to reflect the current scene configuration.

The POV-SLAM system builds upon a recent semi-static map maintenance framework, POCD [6], and the SOTA feature-based RGB-D SLAM system, ORB-SLAM3 [5]. A flow diagram of our novel POV-SLAM system is shown in Figure 2, which consists of five main stages. The following subsections provide an overview of each of the major components in the POV-SLAM pipeline.

We make the following assumptions in this work:

1) The robot operates in a bounded indoor environment (e.g., warehouse or mall) where rigid objects are present.
2) High-level prior knowledge of the objects is available, such as their semantic class, dimension, and likelihood of change.
3) Objects can be added, removed, or shifted between robot traversals, though part of the environment should remain unchanged and observed by the robot.
4) The robot starts its trajectory from a known pose.

B. SLAM Pipeline and Object Representation

The POV-SLAM pipeline takes in a sequence of color and depth frames, $\mathcal{F} = \{\mathbf{F}_t\}_{t=1\dots T}$, from a RGB-D camera, $\mathcal{C}$, as inputs at timestamps $t \in \{1 \dots T\}$. The pipeline outputs the 6-DoF world-to-camera transformations, $\mathcal{T}^{CW} = \{\mathbf{T}_t^{CW} = \{\mathbf{p}_t^{CW}, \mathbf{q}_t^{CW}\}\}_{t=1\dots T}$, with 3D position, $\mathbf{p}_t^{CW}$, and orientation, $\mathbf{q}_t^{CW}$, at each timestep t, along with a library of mapped objects, $\mathcal{O} = \{\mathbf{O}_i\}_{i=1\dots I}$. Each object, $\mathbf{O}_i$, consists of:

- a 4-DoF global pose, $\mathbf{T}_i^{OW} = \{\mathbf{p}_i^{OW}, \phi_i^{OW}\}$,
- a point cloud from accumulated depth data, $\mathbf{P}_i$, and the resulting TSDF reconstruction, $\mathbf{M}_i$,
- a bounding box, $\mathbf{B}_i$, aligned with the major and minor axes of the object reconstruction,

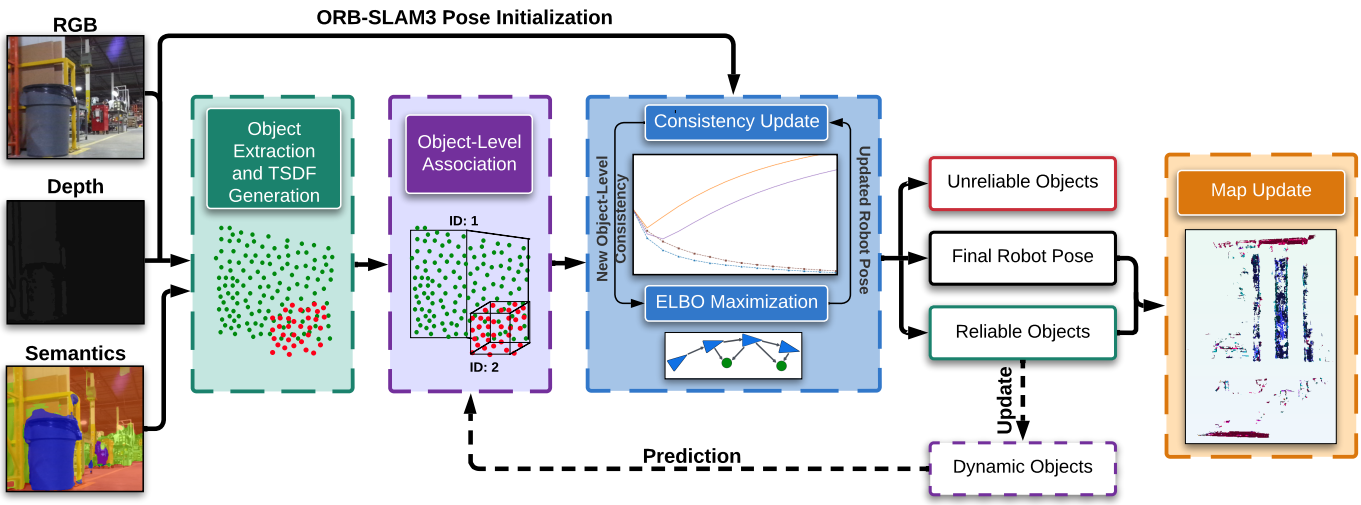

Fig. 2: Our object-aware simultaneous localization and mapping framework (Section III) for semi-static environments. The system inputs are semantically annotated RGB-D frames (Section III-B). Object point clouds are first extracted and TSDFs are generated. Current object TSDFs are then associated with those stored in the object library (Section III-C). An object-level probabilistic consistency update and an evidence lower bound (ELBO) maximization (Section III-D) are performed iteratively to estimate the state of each object and localize the robot (Section III-E). Upon convergence, the outputs are: unreliable objects, which are discarded, reliable objects, and the final robot pose, which are used to generate and update the dense scene map (Section III-F). The framework can be extended to additionally handle dynamic objects (dotted lines) by following MOT-based approaches [14, 31].

- a semantic class, c_i,
- a state probability distribution, $p(l_i, v_i)$, to model the object-level geometric change, $l_i \in \mathbb{R}$, and the consistency, $v_i \in [0, 1]$.
- a set of associated 3D landmark points in the world frame, $\mathcal{L}_i^W = \{\mathbf{l}_{i,l}^W \in \mathbb{R}^3\}_{l=1\ldots L}$,
- the relative positions of the landmark points with respect to the object pose, $\mathbf{T}_i^{OW}$, $\mathcal{L}_i^O = \{\mathbf{l}_{i,l}^O \in \mathbb{R}^3\}_{l=1\ldots L}$,

As we consider indoor mobile robot applications, objects are restricted to only rotate around the z-axis, resulting in a 4-DoF pose, although extending to 6-DoF is trivial. The SLAM system is initialized with an empty object library, $\mathcal{O} = \varnothing$. Along with the camera pose and object models, the system also maintains a dense TSDF map which can be used for downstream tasks such as perception-based planning and control [39, 40].

C. 3D Observation Extraction and Data Association

When a new RGB-D frame, $\mathbf{F}_t$, is received by the system, a set of 3D observations, $\mathcal{Y}_t = \{\mathbf{Y}_{t,j}\}_{j=1\ldots J}$, is extracted and associated to the mapped objects by following the POCD semantic-geometric clustering and association strategy [6]. Additionally, each observation, $\mathbf{Y}_{t,j}$, contains the unprojected 3D keypoints, $\mathcal{D}_{t,j}^C = \{\mathbf{d}_{t,j,d}^C \in \mathbb{R}^3\}_{d=1\ldots D}$, detected from the masked color image. For each associated object-observation pair, $\{\mathbf{O}_i, \mathbf{Y}_{t,j}\}$, we also match the unprojected keypoints, $\mathcal{D}_{t,j}^C$, to the object landmark points, $\mathcal{L}_i^C$.

D. Object Consistency-Augmented Factor Graph

In POCD [6] the authors introduced a Bayesian update rule to propagate an object-level state model, $p(l, v)$. This model consists of a Gaussian distribution which captures the magnitude of the object-level geometric change, l, and a Beta distribution which estimates the consistency between the incoming measurement and the previously mapped object, v. In this work, we exploit the Beta parametrization of consistency $p(v) := \text{Beta}(v \mid \alpha, \beta)$, to estimate the reliability of the object observations in a factor graph optimization framework.

We first consider a simple sparse SLAM problem in a semi-static scene, where previously mapped objects are either moved or unchanged when the robot revisits the region. Our goal is to estimate the robot trajectory, $\mathcal{T}^{CW}$, and determine which of the objects have changed. Existing methods such as ORB-SLAM3 [5] wrap landmark measurement residuals with a robust kernel (e.g., Cauchy loss function) and run optimization multiple times to reject outlier measurements. However, such approaches are not robust to large changes in the scene. Instead, similar to [36, 37], we augment the joint likelihood of our sliding window estimation problem with the object-level Beta-parametrized consistencies, $\{p(v_i)\}_{i=1\ldots I}$, to explicitly model the reliability of each observed landmark:

$$\log p(\mathcal{O}, \{\mathbf{T}_t^{CW}\}_{t=T-m\ldots T}, \{\mathcal{Y}_t\}_{t=T-m\ldots T}) \tag{1a}$$
$$\propto \sum_t \log p(\mathbf{e}_t^{\text{pose}}) \tag{1b}$$
$$+ \sum_i \sum_l \log p(\mathbf{e}_{i,l}^{\text{rigid}}) \tag{1c}$$
$$+ \sum_i \sum_l \log p(\mathbf{e}_{i,l}^{\text{prior}}) \tag{1d}$$
$$+ \sum_t \sum_j \sum_d \log p(\mathbf{e}_{t,j,d}^{\text{key-pt}}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \tag{1e}$$

The factor in Equation (1b) is the transition model. We use the ORB-SLAM3 RGB-D front-end to obtain a visual

odometry (VO) measurement in the body frame, $\mathbf{T}^C_{t-1,t}$, which is used as a prior to initialize the augmented factor graph:

$$\begin{aligned} p(\mathbf{e}^{\text{pose}}_t) &= \mathcal{N}(\mathbf{e}^{\text{pose}}_t \mid \mathbf{0}, \sigma^2_{\text{pose}}\mathbf{I}) \\ \mathbf{e}^{\text{pose}}_t &= (\mathbf{T}^{CW}_{t-1}\mathbf{T}^{CW-1}_t)^{-1}\mathbf{T}^C_{t-1,t} \end{aligned} \tag{2}$$

This factor minimizes the deviation between the estimated relative pose in the body frame and the VO measurement. In practice, we find that this factor improves the stability of the nonlinear optimization.

The factor in Equation (1c) constrains the relative positions of associated object landmarks with respect to the object frame to penalize the deformation of the object geometry:

$$\begin{aligned} p(\mathbf{e}^{\text{rigid}}_{i,l}) &= \mathcal{N}(\mathbf{e}^{\text{rigid}}_{i,l} \mid \mathbf{0}, \sigma^2_{\text{rigid}}\mathbf{I}) \\ \mathbf{e}^{\text{rigid}}_{i,l} &= \mathbf{T}^{OW}_i \mathbf{l}^W_{i,l} - \mathbf{l}^O_{i,l} \end{aligned} \tag{3}$$

The factor in Equation (1d) encourages landmark points to remain at their original positions during optimization. This is important, as objects that have changed but not been rejected can lead to localization errors and a corrupted map, especially at early stages of the optimization process:

$$\begin{aligned} p(\mathbf{e}^{\text{prior}}_{i,l}) &= \mathcal{N}(\mathbf{e}^{\text{prior}}_{i,l} \mid \mathbf{0}, \sigma^2_{\text{prior}}\mathbf{I}) \\ \mathbf{e}^{\text{prior}}_{i,l} &= \mathbf{l}^W_{i,l} - \mathbf{l}^W_{i,l,\text{prev}} \end{aligned} \tag{4}$$

Note that, for simplicity, we use Gaussian measurement likelihoods and isotropic covariance with magnitude $\sigma^2_{\text{pose}}, \sigma^2_{\text{rigid}}, \sigma^2_{\text{prior}}$ for these three factors.

The factor in Equation (1e), the landmark measurement model between an object landmark point, $\mathbf{l}^W_{i,l}$ and its observation, $\mathbf{d}^C_{t,j,d}$, is more complicated, as a Gaussian likelihood is not sufficient to model possible changes in a semi-static scene. An intuitive approximation is to adopt the same Gaussian-Uniform mixture, weighted by the expectation of the Beta consistency model, $\mathbb{E}[v]$, as in [6]:

$$\begin{aligned} p(\mathbf{e}^{\text{key-pt}}_{t,j,d}) &= \mathbb{E}[v]\mathcal{N}(\mathbf{e}^{\text{key-pt}}_{t,j,d} \mid \mathbf{0}, \sigma^2_{\text{key-pt}}\mathbf{I}) \\ &\quad + (1-\mathbb{E}[v])\mathcal{U}(\|\mathbf{e}^{\text{key-pt}}_{t,j,d}\|_2 \mid 0, e_{\max}) \\ \mathbf{e}^{\text{key-pt}}_{t,j,d} &= \mathbf{T}^{CW-1}_t \mathbf{d}^C_{t,j,d} - \mathbf{l}^W_{i,l} \end{aligned} \tag{5}$$

This mixture model consists of two parts: 1) a zero-mean Gaussian component with an isotropic measurement covariance, $\sigma^2_{\text{key-pt}}$, for the unchanged scenario, and 2) a uniform component with a predefined maximum association distance, $e_{\max}$, for the changed scenario in which the object could be anywhere. However, using the single point estimator, $\mathbb{E}[v]$, could lead to an inaccurate estimation as it does not capture the full Beta consistency distribution. We present a variational formulation to derive an ELBO for the landmark measurement model in Section IV-A, which is efficient to implement, and shown to provide better convergence behavior than the single point approximation in Equation (5). Figure 3 illustrates the complete factor graph.

Optionally, our framework can be extended to handle dynamic objects in the scene. We follow the strategy in [31], where the poses and associated landmarks of moving objects

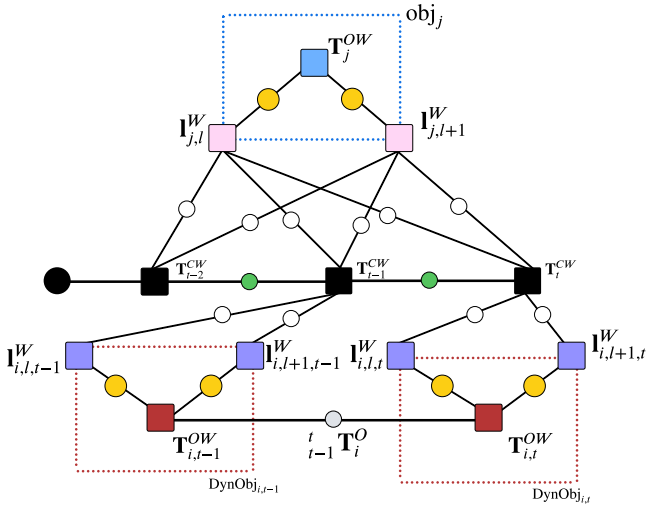

Fig. 3: A factor graph representation of our probabilistic, object-aware SLAM method, solved in the M-step of each EM iteration (Section IV-C), with **blue** semi-static and **red** optional dynamic objects. **Black squares**: robot poses at each time step, **black circle**: robot prior factor, **green circles**: odometry factors, **white circles**: ELBO of object landmark measurement factors (Section IV-A), **pink** and **purple**: landmarks associated with static and dynamic objects, respectively, **orange circles**: object rigidity priors, **grey circle**: optional dynamic object velocity factor.

are modeled at each timestamp in the window, temporally constrained by the estimated velocity. This strategy is tested in simulation, as discussed in the Supplementary Material.

E. Iterative Object Consistency Update and Pose Estimation

The augmented optimization problem in Equation (1) consists of both unknown parameters, which are the robot trajectory and the object poses with their associated landmark positions, and a set of unobserved latent variables, which are the object consistencies. Although the problem is complex to solve directly, a favoured approach to solving such estimation problems involving latent variables is iteratively via expectation-maximization (EM). We introduce an EM-based method in Section IV, which leverages our variational landmark measurement model to solve the factor graph iteratively at every frame, $\mathbf{F}_t$.

F. Object and TSDF Map Update

Once optimization is complete, we extract the new robot and object pose information, and update the map and object library. All new object observations not associated to the previous map are integrated into the TSDF map and added to the object library, $\mathcal{O}$. A large pseudo-change is used to penalize the consistency of objects currently in the camera frustum, but not associated with any observations. Objects *accepted* by the VEM optimization, as discussed in Section IV-C, are considered consistent with the map, and their observations are integrated into the object's TSDF model, M_i. Their object state models are then propagated by one step based on the new robot pose estimate. The states of objects not accepted by the VEM

optimization are also propagated by one step, though their observations are not integrated, as they are no longer consistent with their previous models. If an object's consistency expectation, $\mathbb{E}[v_i]$, falls below a pre-defined threshold, θ_{consist}, the object is removed from the library, and all associated voxels in the TSDF model are reinitialized. Note that the rejected objects are not discarded immediately after optimization to ensure robustness against potential measurement noise and pose estimation error in the current frame. When dynamic objects are considered, we can update their motion models with their new pose estimates using a Kalman filter.

IV. Methodology

In this section, we discuss the details of our VEM method: 1) In the E-step we compute the ELBO for the expectation of the landmark measurement likelihood for potentially changing objects, and 2) in the M-step we optimize the approximated factor graph to update the robot and object states. Algorithm 1 in the Supplementary Material outlines how our pipeline processes one frame to update the robot and object states.

A. E-Step: ELBO of Measurement Likelihood for Potentially Semi-Static Objects

In Section III-D, the cost function for the augmented factor graph SLAM problem is introduced, where each object and its associated landmark points share a Beta-parametrized consistency estimate. Such a problem is challenging to optimize, even with the EM algorithm. Moreover, as discussed earlier, a single point approximation using the consistency expectation, $\mathbb{E}[v]$ (Equation (5)), does not capture the full Beta consistency model. In this section, we focus on the E-Step of the VEM algorithm and derive the ELBO for the expectation of each object landmark's measurement likelihood in Equation (1), based on the robot trajectory, object landmark position, and object consistency estimated in the previous EM iteration.

Consider a single landmark from an object. At frame T and EM iteration n, we obtain a Beta consistency posterior, $\text{Beta}(\alpha, \beta)$, for the object by following the Bayesian method introduced in [6], with respect to the current frame measurement, $\mathbf{d}_T^C$, the previous iteration's landmark position estimate, $\mathbf{l}^W$, and robot pose estimate, $\mathbf{T}^{CW}$. Note that these variables are treated as constants in the E-Step. The timestamps and indices in the notation are dropped for clarity. The object's true consistency, $\pi \in \{0, 1\}$, can be considered as a sample from a Bernoulli distribution parametrized by v, with $\pi = 1$ indicating the object has not changed. We can then write a generative process, $p(\pi, v) = p(\pi \mid v)p(v \mid \alpha, \beta)$, where:

$$\begin{aligned} v &\sim \text{Beta}(\alpha, \beta) \\ \pi &\sim \text{Bernoulli}(v) \end{aligned} \tag{6}$$

Unchanged objects will follow a zero-mean, isotropic Gaussian measurement model and moved objects can be anywhere in the scene. The measurement residual, $\mathbf{e}_T$, is defined to be the 3D point-wise distance:

$$\mathbf{e}_T = \mathbf{T}^{CW-1}\mathbf{d}_T^C - \mathbf{l}^W \tag{7}$$

We can then rewrite the Gaussian-Uniform measurement model weighted by the sampled object consistency, π, as:

$$\begin{aligned} p(\mathbf{e}_T) &:= p(\mathbf{e}_T \mid \mathbf{T}^{CW}, \mathbf{l}^W, \pi) \\ &= \mathcal{N}(\mathbf{e}_T \mid \mathbf{0}, \sigma^2\mathbf{I})^{\pi}\mathcal{U}(\|\mathbf{e}_T\|_2 \mid 0, e_{\text{max}})^{1-\pi} \end{aligned} \tag{8}$$

Since π is sampled from the generative process shown in Equation (6), Equation (8) involving dependent latent variables, $\boldsymbol{\omega} = \{\pi, v\}$, is challenging to maximize. Fortunately, we can apply the mean field approximation [41] by assuming the two latent variables are fully independent, $p(\pi, v) \simeq q(\pi)q(v)$. This would allow us to write a variational lower bound, $\mathcal{L}$, for the evidence, $\log p(\mathbf{e}_T \mid \mathbf{T}^{CW}, \mathbf{l}^W, \alpha, \beta)$:

$$\mathcal{L}(\boldsymbol{\omega}, \mathbf{T}^{CW}, \mathbf{l}^W) = \mathbb{E}_{q(\boldsymbol{\omega})}\left[\log \frac{p(\mathbf{e}_T, \boldsymbol{\omega} \mid \mathbf{T}^{CW}, \mathbf{l}^W, \alpha, \beta)}{q(\boldsymbol{\omega})}\right] \tag{9}$$

where the joint likelihood is

$$\begin{aligned} &\log p(\mathbf{e}_T, \boldsymbol{\omega} \mid \mathbf{T}^{CW}, \mathbf{l}^W, \alpha, \beta) \\ &= \log p(\mathbf{e}_T \mid \mathbf{T}^{CW}, \mathbf{l}^W, \pi) + \log p(\pi \mid v) + \log p(v \mid \alpha, \beta) \\ &= \pi[\log v + \log \mathcal{N}(\mathbf{e}_T \mid \mathbf{0}, \sigma^2\mathbf{I})] \\ &\quad + (1-\pi)[\log(1-v) + \log \mathcal{U}(\|\mathbf{e}_T\|_2 \mid 0, e_{\text{max}})] \\ &\quad + \log \text{Beta}(v \mid \alpha, \beta) \end{aligned} \tag{10}$$

Following the mean field approximation, the optimal $q(\pi)$ and $q(v)$ that maximize the lower bound (9) are:

$$\begin{aligned} \log q(\pi) =& \pi\big[\mathbb{E}[\log v] + \log \mathcal{N}(\mathbf{e}_T \mid \mathbf{0}, \sigma^2\mathbf{I})\big] \\ &+ (1-\pi)\big[\mathbb{E}[\log(1-v)] \\ &\quad + \log \mathcal{U}(\|\mathbf{e}_T\|_2 \mid 0, e_{\text{max}})\big] + const \\ \log q(v) =& \mathbb{E}[\pi] \log v + \mathbb{E}[1-\pi] \log(1-v) \\ &+ \log \text{Beta}(v \mid \alpha, \beta) + const \end{aligned} \tag{11}$$

Now, the expectation of the probability that the object did not change, $\mathbb{E}[\pi]$, can be computed based on the current measurement and estimates:

$$\begin{aligned} \mathbb{E}[\pi] &= q(\pi = 1) \\ &= \eta \exp\{\mathbb{E}[\log v] + \log \mathcal{N}(\mathbf{e}_T \mid \mathbf{0}, \sigma^2\mathbf{I})\} \\ \mathbb{E}[1-\pi] &= q(\pi = 0) \\ &= \eta \exp\{\mathbb{E}[\log(1-v)] + \log \mathcal{U}(\|\mathbf{e}_T\|_2 \mid 0, e_{\text{max}})\} \end{aligned} \tag{12}$$

Here, η is a normalizing factor, and $\mathbb{E}[\log v]$ and $\mathbb{E}[\log(1-v)]$ can be computed from the property of the Beta distribution:

$$\begin{aligned} \mathbb{E}[\log v] &= \psi(\alpha) - \psi(\alpha + \beta) \\ \mathbb{E}[\log(1-v)] &= \psi(\beta) - \psi(\alpha + \beta) \end{aligned} \tag{13}$$

where $\psi(\cdot)$ is the digamma function. Finally, we can compute the lower bound:

$$\begin{aligned} &\mathcal{L}(v, \pi, \mathbf{T}^{CW}, \mathbf{l}^W) \\ &= \mathbb{E}[\pi] \log \mathcal{N}(\mathbf{e}_T \mid \mathbf{0}, \sigma^2\mathbf{I}) \\ &\quad + \mathbb{E}[1-\pi] \log \mathcal{U}(\|\mathbf{e}_T\|_2 \mid 0, e_{\text{max}}) + const \end{aligned} \tag{14}$$

Comparing to the naive approximation in Equation (5), the ELBO is a mixture between a log-Gaussian mode and a log-Uniform mode. However, the new weights, $\mathbb{E}[\pi]$ and $\mathbb{E}[1-\pi]$, incorporate the full Beta consistency model as well as the likelihood of the two modes. This provides a more statistically consistent measurement model for potentially changing objects. We refer the reader to the Supplementary Material for a more detailed derivation, as well as a performance comparison against the single point approximation in Equation (5).

B. ELBO Tightness and Assumptions

We repeat the ELBO estimation (Equation (14)) presented in Section IV-A for all observed objects and their landmarks in the scene, which we substitute into the joint likelihood, discussed in Section III-D, to construct a lower bound to the original optimization cost (Equation (1)) for our sliding window SLAM problem. The new factor graph can be solved efficiently using an available SLAM solver, such as *g2o* [42].

Unfortunately, sub-optimal or diverged solutions are likely to occur. The mean field approximation used in the measurement ELBO tends to be overconfident [41], especially when the Beta consistency estimate is uncertain. On the other hand, the ELBO tightens when the Beta distribution approaches a unit impulse, i.e., when $\alpha \gg \beta$ or $\alpha \ll \beta$. This implies that when object consistency estimates are uncertain, the lower bound can be improved but there is no guarantee to improve the true joint likelihood, as some moved objects can be misclassified as unchanged. Nonetheless, with additional iterations the ELBO tightens, improving the true likelihood. This convergence behavior requires that: 1) a good prior robot pose is available, and 2) some distinguishable, unchanged objects are observed by the robot. We believe these are reasonable assumptions to make in the semi-static SLAM problem. Most robots deployed in industrial settings depart from and return to pre-determined charging stations. Visual place recognition techniques can also be used to initialize the system. Moreover, if the robot only observes changed objects, then it is not possible to determine the global pose of the robot just using vision data. Without inertia or off-board anchor sensors (e.g., IMUs and UWBs), the system will converge to a minimum-cost state but there is no guarantee to the correctness. We provide simulation results in the Supplementary Material to illustrate the system's behavior under advertorial scenarios.

C. M-Step: Factor Graph Optimization

In order to exploit the aforementioned assumptions and encourage the optimizer to make use of static objects with higher certainty to perform system updates, a max-mixture [43] approach is adopted to guide the optimization process. At every gradient descent step, for every object landmark, a weighted log measurement likelihood is computed for the unchanged and moved scenarios, and a decision is made on whether the measurement should be *accepted* in computing the gradient:

$$m = \operatorname{argmax}\{\log \mathbb{E}[v]\mathcal{N}(\mathbf{e}_T \mid \mathbf{0}, \sigma^2\mathbf{I}), \log(1-\mathbb{E}[v])\mathcal{U}(\|\mathbf{e}_T\|_2 \mid 0, e_{\max})\} \tag{15}$$

$$\tilde{\mathcal{L}}(v, \pi, \mathbf{T}^{CW}, \mathbf{l}^W) := \begin{cases} \mathbb{E}[\pi]\log\mathcal{N}(\mathbf{e}_T \mid \mathbf{0}, \sigma^2\mathbf{I}), & \text{if } m = 0 \\ \mathbb{E}[1-\pi]\log\mathcal{U}(\|\mathbf{e}_T\|_2 \mid 0, e_{\max}), & \text{if } m = 1 \end{cases} \tag{16}$$

This approximation excludes objects with lower measurement likelihood from contributing to the overall cost, achieving faster and more accurate convergence when the ELBOs are not tight. Rejected objects are not deleted immediately, but revised at every gradient step. Note that we choose $\mathbb{E}[v]$ instead of $\mathbb{E}[\pi]$ to weight the measurement likelihoods when making the rejection decisions. Empirical results show that $\mathbb{E}[\pi]$, despite being a more accurate estimate, could be highly noisy due to the overconfidence in the mean field approximation. On the other hand, $\mathbb{E}[v]$ comes from the Bayesian update rule, thus providing smoother gradients to achieve more stable convergence. More details and ablation studies are provided in the Supplementary Material.

Substituting the approximated ELBO (Equation (16)) into Equation (1), and maximizing the new factor graph, we obtain the updated robot and object states for the next EM iteration:

$$\begin{aligned} \mathcal{O}, \mathcal{T}^{CW} = & \\ & \underset{\mathcal{O}, \mathcal{T}^{CW}}{\operatorname{argmax}} \log p(\mathcal{O}, \mathcal{T}^{CW}, \{\mathcal{Y}_t\}_t) \\ & \propto \sum_t \log p(\mathbf{e}_t^{\text{pose}}) \\ & + \sum_i \sum_l \log p(\mathbf{e}_{i,l}^{\text{rigid}}) \\ & + \sum_i \sum_l \log p(\mathbf{e}_{i,l}^{\text{prior}}) \\ & + \sum_t \sum_i \sum_l \tilde{\mathcal{L}}(v_i, \pi_i, \mathbf{T}_t^{CW}, \mathbf{l}_l^W) \end{aligned} \tag{17}$$

Our VEM formulation ensures a monotonically increasing ELBO until a zero-gradient solution but does not guarantee convergence to an optimum. Global optimality is inherently challenging, but our descent method mostly provides high-quality solutions when assumptions in Section IV-B are met.

V. Experimental Results

A. Experimental Setup

We verify the performance of our framework qualitatively and quantitatively by comparing both the map reconstruction and robot trajectory error of POV-SLAM to:

- ORB-SLAM3 [5]: A SOTA sparse visual SLAM solution, which assumes the world is static.
- VI-MID [17]: A recent object-level SLAM method for small (5m×5m) semi-static scenes. The method performs dense RGB-D tracking on certainly-static regions based on semantics for camera localization, before updating the object states. As the code was unavailable, we modify ORB-SLAM3 to exclude features from potentially changing objects during pose estimation, and use POCD [6] for object change detection and mapping. Our custom implementation is referred to as "Ours-MID".

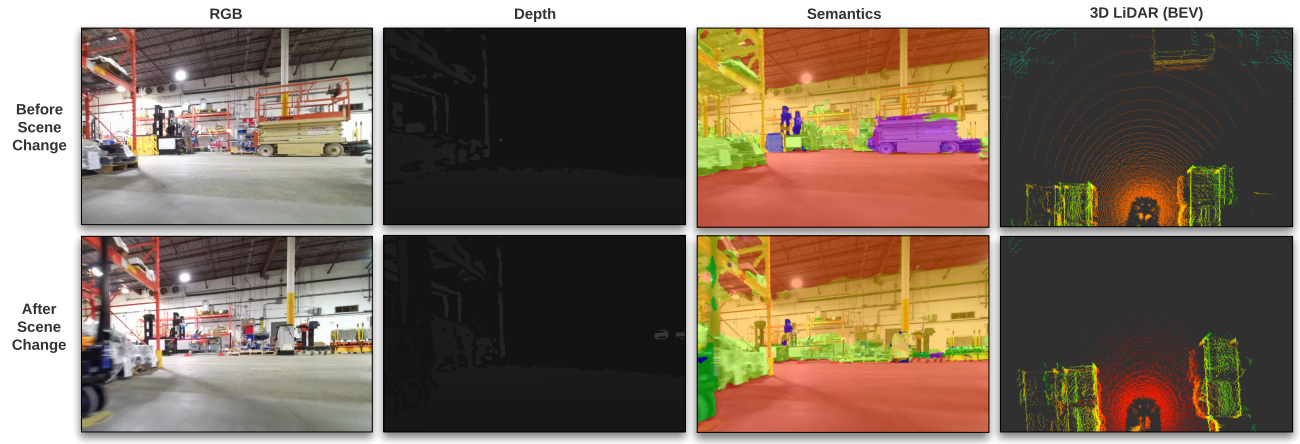

Fig. 4: A sample RGBD, semantically labelled, and 3D LiDAR frame captured by the sensors before **(top)** and after **(bottom)** the scene changes. The forklift in the top frame, captured June 15, 2022, is no longer present in the bottom frame, captured Oct. 23, 2022.

In essence, ORB-SLAM3 uses all features, Ours-MID employs certainly-static features, and POV-SLAM probabilistically selects features from likely-unchanged objects. In a static scene, POV-SLAM should revert to regular batch SLAM where only the Gaussian mode of the measurement model is active. To demonstrate the capabilities of POV-SLAM, we evaluate in three scenarios: 1) a 2D simulation (Section V-C), 2) a 3D synthetic semi-static dataset (Section V-D), and 3) our real-world, semi-static warehouse dataset (Section V-E). The lack of large, real-world SLAM datasets with multiple passes through environments that include both dynamic and semi-static objects prompted us to create one (Section V-B). We implement our method on top of ORB-SLAM3 [5] and POCD [6]. The parameters used to evaluate against all methods can be found in the Supplementary Material.

To benchmark 3D reconstruction accuracy under scene changes, we generate the ground truth meshes by using POCD with the ground truth robot trajectory. As POCD has shown to outperform several mapping methods (Kimera [3], Fehr *et al.* [18], and Panoptic Multi-TSDFs [19]), the mesh obtained is representative of the best possible reconstruction.

Note that in this work, we use RGB-D information to address scene changes directly, thus inertial and odometry data are excluded. While IMUs can supplement all methods, our method yields RGB-D pose estimates that align better with IMU data, removing errors at their source.

B. Real-World Semi-Static Warehouse Dataset

We release an extension to the TorWIC change detection dataset [6]. The original TorWIC dataset features a small 10m$\times$10m hallway setup using boxes and fences with limited real-world objects and changes. Its ground-truth trajectory, acquired via 2D LiDAR SLAM, suffers from jumps and drifts and thus not suitable for evaluating SLAM algorithms. Conversely, the new extension, as the first long-term real-world warehouse dataset, originates from an active 100m$\times$80m Clearpath Robotics plant showcasing various objects and changes (e.g., forklifts, robots, people).

The dataset is collected on a mobile base equipped with two Microsoft Azure RGB-D cameras, an Ouster 128-beam LiDAR, and two IMUs. We repeat three scenarios over the course of four months, presenting changed object locations over time, with a total of 20 trajectories. The robot setup, sensor specifications and the scenario breakdown can be found in the Supplementary Material. Figure 4 shows the scenario changes for a sample route.

To facilitate SLAM and reconstruction evaluation, we also release the ground truth scan of the warehouse and ground truth trajectories. A Leica MS60 multistation was used to obtain a centimetre-level accurate point cloud of the warehouse. Iterative closest point (ICP) was performed between the 128-beam LiDAR scan and the ground truth scan to obtain highly accurate ground truth trajectories for the robot. The robot starts and ends at the pre-defined map origin, so users can easily stitch trajectories to create long routes with change.

C. 2D Semi-Static Simulation

In this section, we introduce the first of three experiments performed. A 2D simulation was constructed to demonstrate our probabilistic framework in action, and to justify the design choices made. The setup can be seen in Figure 5, consisting of four unchanged objects and six moved objects. The robot is spawned around its ground truth pose, with noise in both position and orientation, and drives in the scene. The robot measures the four vertices of the rectangular objects, all corrupted by Gaussian noise. As seen in Figure 5, the system is able to correctly identify the six moved boxes, recovering their true poses. In the Supplementary Material, the evolution of the state estimates of the system over the first four frames are shown, as the robot navigates the scene.

Figure 6 shows the evolution of the object consistency expectation, $\mathbb{E}[v]$, and the robot pose error over the EM iterations at the first frame. The consistency expectations converge to their true values at the end of the optimization and the robot pose converges to its ground truth after six EM iterations. There is a drop in the consistency of all objects during the first iterations due to the initial error in the robot

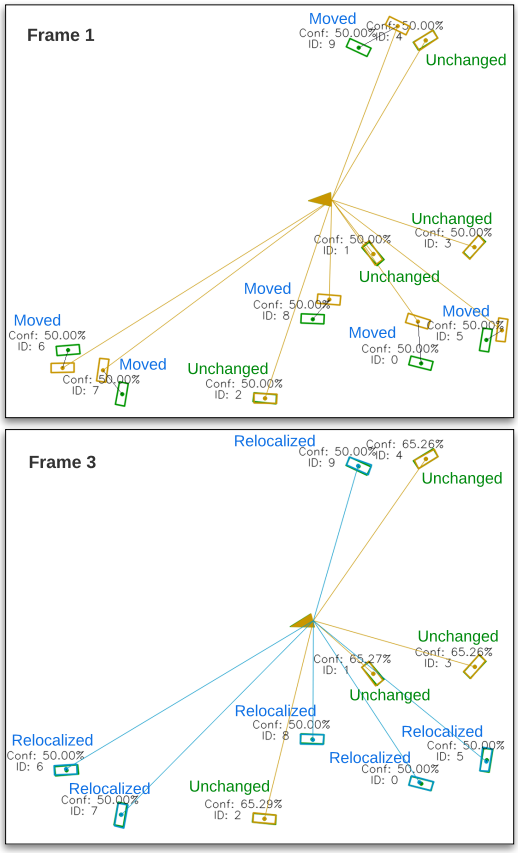

Fig. 5: The setup of our 2D semi-static simulation with six moved objects and four unchanged objects over the first and third frames of optimization. **Rectangle**: object, **triangle**: robot, **green**: ground truth, **yellow**: estimation, **blue**: reconstructed objects. All object states are initialized with a $\mathbb{E}[v] = 0.5$ consistency estimate. All moved boxes are identified and relocalized after three frames.

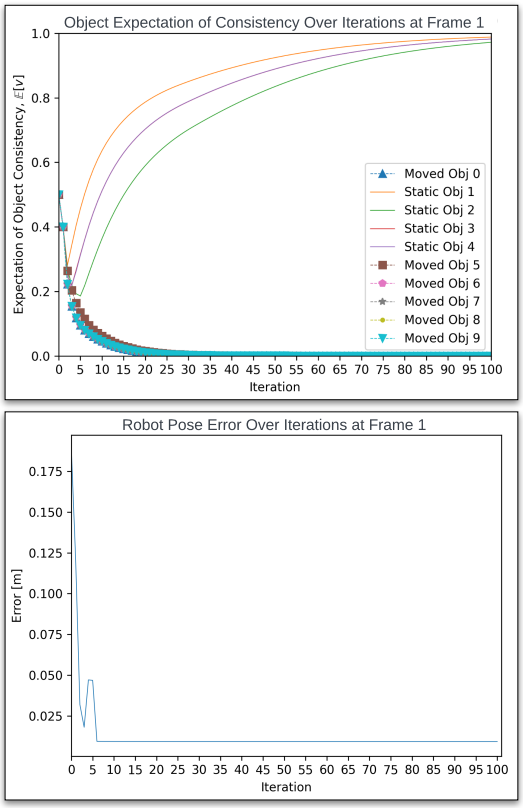

Fig. 6: The object-level consistency expectation (top) and robot pose error (bottom) over VEM iterations at the first frame for the simulated scenario. For the object consistencies, plots with markers belong to moved objects. Although the consistencies converge after 100 iterations, in practice we can stop much earlier.

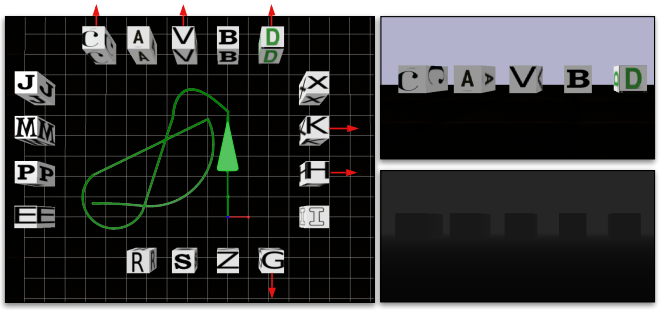

Fig. 7: The setup for the 3D simulation (left). **Green**: robot trajectory, **red**: direction of box motion. The RGB (top) and depth (bottom) camera views can be seen on the right.

pose estimate. However, as the robot pose becomes more accurate, the true states are recovered. This experiment shows the robustness of our method, as the system is able to recover the true state even when the number of moved objects in the scene exceeds the number of unchanged objects.

We shall note that the iterative optimization process finds the most likely underlying scene configuration based on the measurements. Therefore, if there exists a different hypothesis that exhibits a higher measurement likelihood, the optimizer would converge to that solution. For example, if the six moved objects had all shifted in the same direction by the same magnitude, our system would mark them as stationary and relocalize the unchanged objects instead. However, since such scenarios cannot be distinguished from a probabilistic point of view, they are not of concern. This adversarial scenario is shown in the Supplementary Material.

Further ablation studies showcasing the advantage of using the ELBO instead of the single point estimate (Section III-D), the use of max-mixture approximation (Section IV-C), the choice of weights ($\mathbb{E}[v]$ vs $\mathbb{E}[\pi]$) to use when choosing the mode of max-mixture (Section IV-C), and an adversarial fully dynamic scenario, are available in the Supplementary Material.

D. 3D Semi-Static Simulation

In this section, we introduce a 3D simulated semi-static scene, henceforth referred to as "BoxSim". The setup can be seen in Figure 7. The robot moves among 17 boxes, six of which shift between robot traversals. The scenario is very challenging as there is no static background available, requiring all methods to localize against the 17 boxes.

Figure 1 visually compares the robot trajectories estimated by ORB-SLAM3 and POV-SLAM against the ground truth.

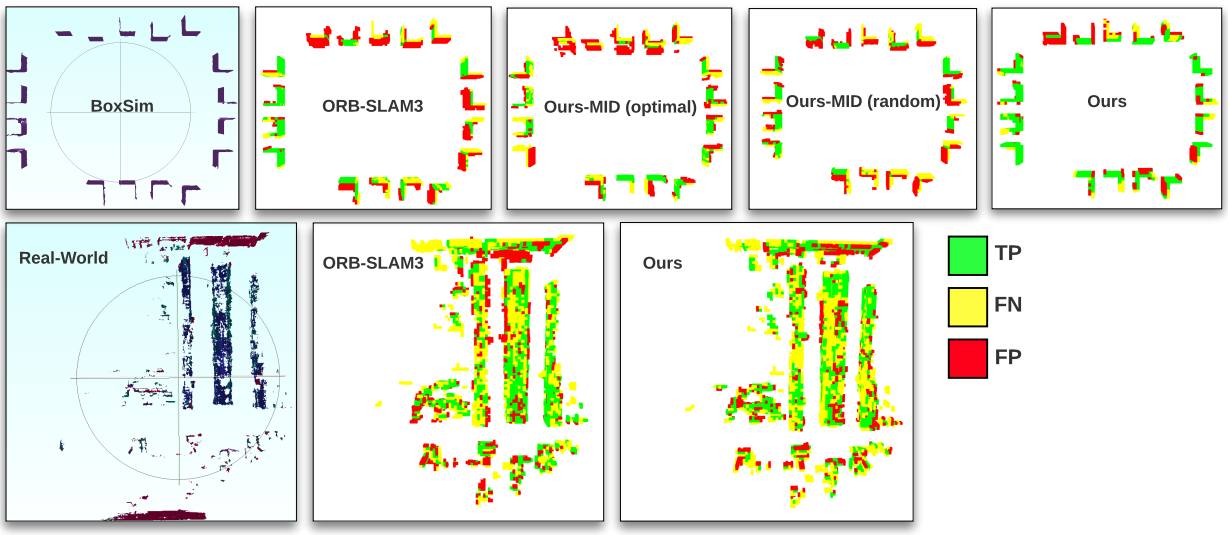

Fig. 8: A bird's-eye-view qualitative analysis of 3D reconstruction results of the **top row:** BoxSim synthetic dataset compared against that of ORB-SLAM3 and Ours-MID, and **bottom row:** our real-world warehouse dataset, compared against that of ORB-SLAM3. The green, yellow, and red sections represent true positives (correct prediction), false negatives (incorrect negative prediction), and false positives (incorrect prediction), respectively. The first image is the ground truth map of the routes' final scenario after scene change. For BoxSim a grid of 0.2 and for the real-world trajectory a grid size of 0.4 were used to voxelize the reconstruction.

As discussed, ORB-SLAM3 assumes a static environment. Although it utilizes robust kernels and iterative pruning to reject outlier landmarks, it is still sensitive to large scene change. As seen in the figure, its estimated trajectory diverges from the ground truth when changed objects are encountered. On the other hand, POV-SLAM optimizes a lower bound to a Gaussian-Uniform likelihood to explicitly infer if any of the mapped objects have changed, resulting in much smoother and accurate pose estimates.

As discussed in the literature review, a common method for dynamic object handling is to ignore all potentially moving objects. In VI-MID [17] the authors mask out all potentially-changing objects based on semantic information, performing dense tracking on the static background alone. However, this relies on the assumption that the changing parts of the environment are known, which is not feasible in the real world. We evaluate our adaptation, Ours-MID, on two scenarios: 1) the **optimal** case, where the system knows which objects will shift and 2) the **random** case, where objects are randomly chosen to represent the static background.

TABLE I: Absolute Trajectory Error (ATE) and Maximum Position Error (MPE) on the BoxSim Dataset.

BoxSim	ATE [m]	MPE [m]
ORB-SLAM3	0.14	0.47
Ours-MID (optimal)	0.41	0.82
Ours-MID (random)	0.49	0.90
POV-SLAM (ours)	**0.10**	**0.26**
POV-SLAM Improvement	0.04 (29%)	0.21 (45%)

TABLE II: Quantitative mapping results on the BoxSim Dataset.

BoxSim	Precision ↑	Recall (TPR) ↑	FPR ↓
ORB-SLAM3	52.1	72.4	3.8
Ours-MID (optimal)	45.2	56.7	4.1
Ours-MID (random)	39.0	50.5	4.6
POV-SLAM (ours)	**68.5**	**78.7**	**2.2**
POV-SLAM Improvement	16.4 (31%)	6.3 (8.7%)	-1.6 (42%)

The average trajectory error (ATE) and maximum position error (MPE) can be seen in Table I. POV-SLAM significantly outperforms ORB-SLAM3 and Ours-MID. For Ours-MID, in the **optimal** case, by ignoring all potentially changing objects the system might not observe enough features when the robot visits locations where moving objects dominate, causing poor estimates. In the **random** case, its performance further degrades when objects are incorrectly classified.

We further compare the dense reconstructions of POV-SLAM against ORB-SLAM3 and Ours-MID. The top row of Figure 8 shows the qualitative comparisons, where we overlay and voxelize both the reconstructions and the ground truth mesh and colorize the overlapping (inlier) and inconsistent (outlier) voxels. We then compute the precision, recall, and false positive rate (FPR) by counting the voxels for a quantitative evaluation, and Table II lists the quantitative results. ORB-SLAM3 and Ours-MID both generated distorted maps with failed object updates due to localization drift, which led to incorrect data association in object consistency update. On the other hand, POV-SLAM generates the most visually correct map where all moved boxes, except the one at the top left, are relocalized to the new locations. Quantitatively, POV-SLAM exhibits the highest precision (coverage of true objects), and the lowest FPR (map update quality after scene change) due to its superior localization performance.

As all methods use the same POCD [6] framework and parameters to perform map update at every frame, this experiment highlights that 1) explicit reasoning of object consistency is required for localization in semi-static environments, and 2) joint estimation of object consistency and robot localization brings significant advantage in cluttered scenes.

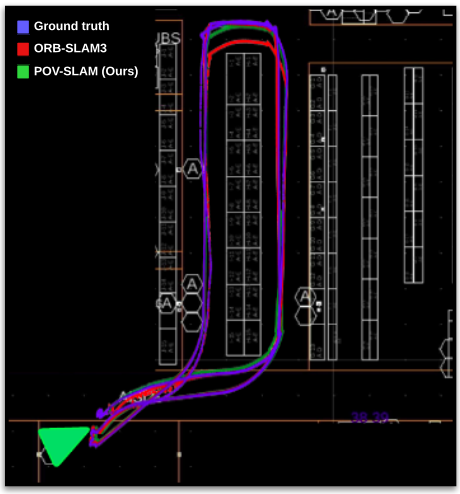

Fig. 9: The output trajectories of ORB-SLAM3 and POV-SLAM through the real-world warehouse aisle scenario.

E. Real-World Experiment in a Semi-Static Scene

In this section, we evaluate POV-SLAM's effectiveness in a warehouse scenario, available through our novel real-world semi-static dataset. We stitch two trajectories captured along the same route four months apart to introduce scene changes as the robot traverses the warehouse. Figure 9 shows the routes overlaid on the factory's schematic floor plan and Figure 4 shows a sample pair of frames with scene changes. Due to the limited effective range of the Azure RGB-D cameras, we rely on the Ouster Lidar to provide feature depth information when traversing in open areas.

We qualitatively and quantitatively compare the trajectory estimation and scene reconstruction results against ORB-SLAM3. Ours-MID is not included in the comparison as the route is cluttered with pallets and boxes, leaving very limited static background information for Ours-MID to localize against. The output trajectories along with the ground truth are visualized in Figure 9. ORB-SLAM3 successfully completes the first traversal with high accuracy. However, changes along the aisle in the second traversal cause incorrect data association and lead to a shortened trajectory. POV-SLAM performs slightly worse than ORB-SLAM3 in the first traversal. However, in the second traversal, POV-SLAM is able to reject the false positive matches and track with higher accuracy. The ATEs and MPEs from the two traversals can be seen in Table III and Table IV.

The bottom row of Figure 8 visualizes the 3D reconstruction results. Again, we voxelize the reconstructed meshes and count for overlapping and inconsistent voxels to obtain the quantitative evaluations, which are listed in Table V. POV-SLAM outperforms ORB-SLAM3 in both localization accuracy and scene reconstruction on this route when scene changes are encountered as it does not suffer from incorrect loop closures.

TABLE III: Absolute Trajectory Error (ATE) and Maximum Position Error (MPE) in Traversal 1 of the Real-World Aisle Scenario.

Real-World-T1	ATE-T1 [m]	MPE-T1 [m]
ORB-SLAM3	**0.14**	**0.31**
POV-SLAM (ours)	0.26	0.48
POV-SLAM Improvement	-0.12 (-86%)	-0.17 (-55%)

TABLE IV: Absolute Trajectory Error (ATE) and Maximum Position Error (MPE) in Traversal 2 of the Real-World Aisle Scenario.

Real-World-T2	ATE-T2 [m]	MPE-T2 [m]
ORB-SLAM3	0.89	1.55
POV-SLAM (ours)	**0.46**	**0.98**
POV-SLAM Improvement	0.43 (48%)	0.57 (37%)

TABLE V: Quantitative mapping results on the Real World Dataset.

Real World	Precision ↑	Recall (TPR) ↑	FPR ↓
ORB-SLAM3	76.2	**56.9**	1.9
POV-SLAM (ours)	**79.7**	53.7	**1.5**
POV-SLAM Improvement	3.5 (4.6%)	-3.2 (-6.0%)	-0.4 (21%)

F. Run-time Performance

With a max of 4,000 ORB features in each frame, a window size of 8, and 30 EM iterations per frame, POV-SLAM runs at approximately 1Hz on a Linux desktop with an AMD Ryzen R9-5900X CPU at 3.7Hz. To achieve a more realistic run-time, we only execute the VEM optimization every seven frames on the real-world dataset, while relying on ORB-SLAM3 in between. We use a large number of ORB features because the dataset is challenging due to varying lighting conditions, causing even the original ORB-SLAM3 to fail at times with the default 1250 features. As well, our object-aware method requires each object to have sufficient features for tracking and association. We currently use a uniform feature detection approach, so small yet key objects may not get enough features under a lower quota. In practice, POV-SLAM is amenable to online operation in large environments, as change detection and localization correction is not required at every frame. A semantic-aware feature extraction approach could further improve the performance in the future.

VI. Conclusion

In this paper we present POV-SLAM, a novel online, probabilistic object-aware framework to simultaneously estimate the robot pose, and track and update object-level scene changes in a joint optimization-based framework. The POV-SLAM pipeline uses our derived variational expectation maximization strategy to optimize factor graphs accounting for potentially-changing objects. We experimentally verify the robustness of POV-SLAM against state-of-the-art SLAM methods on two datasets, including our novel, real-world, semi-static warehouse dataset that we release with this work. Our system explicitly reasons about object-level stationarity to improve the robustness of localization in slowly varying scenes. Our method outperforms ORB-SLAM3 on average trajectory error by 48% on the real-world dataset and 29% on the 3D synthetic semi-static dataset. As well, POV-SLAM shows a 4.6% improvement on dense reconstruction precision in the large real-world scene and 31% in the smaller synthetic scene.

References

[1] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D. Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 2015.

[2] M. Grinvald, F. Furrer, T. Novkovic, J. J. Chung, C. Cadena, R. Siegwart, and J. Nieto. Volumetric Instance-Aware Semantic Mapping and 3D Object Discovery. *IEEE Robotics and Automation Letters*, 2019.

[3] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[4] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. Elasticfusion: Real-time dense SLAM and light source estimation. *The International Journal of Robotics Research*, (14), 2016.

[5] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *arXiv preprint arXiv:2007.11898*, 2020.

[6] Jingxing Qian, Veronica Chatrath, Jun Yang, James Servos, Angela Schoellig, and Steven L. Waslander. POCD: Probabilistic Object-Level Change Detection and Volumetric Mapping in Semi-Static Scenes. In *2022 Robotics: Science and Systems (RSS)*, 2022.

[7] Irene Ballester, Alejandro Fontán, Javier Civera, Klaus H. Strobl, and Rudolph Triebel. DOT: dynamic object tracking for visual SLAM. *CoRR*, 2020.

[8] Chao Yu, Zuxin Liu, Xinjun Liu, Fugui Xie, Yi Yang, Qi Wei, and Fei Qiao. DS-SLAM: A semantic visual SLAM towards dynamic environments. *CoRR*, 2018.

[9] Xiaoyun Lu, Hu Wang, Shuming Tang, Huimin Huang, and Chuang Li. Dm-slam: Monocular SLAM in dynamic environments. *Applied Sciences*, 2020.

[10] J. McCormac, R. Clark, Michael Bloesch, A. Davison, and Stefan Leutenegger. Fusion++: Volumetric object-level SLAM. *2018 International Conference on 3D Vision (3DV)*, 2018.

[11] Ting Sun, Yuxiang Sun, Ming Liu, and Dit-Yan Yeung. Movable-object-aware visual SLAM via weakly supervised semantic segmentation. *ArXiv*, 2019.

[12] Antoni Rosinol, Marcus Abate, Jingnan Shi, and Luca Carlone. 3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans. In *Robotics: Science and Systems*, 2020.

[13] Martin Rünz and Lourdes de Agapito. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2018.

[14] Ryo Hachiuma, Christian Pirchheim, Dieter Schmalstieg, and H. Saito. Detectfusion: Detecting and segmenting both known and unknown dynamic objects in real-time SLAM. In *BMVC*, 2019.

[15] Binbin Xu, Wenbin Li, Dimos Tzoumanikas, Michael Bloesch, Andrew J. Davison, and Stefan Leutenegger. MID-fusion: Octree-based object-level multi-instance dynamic SLAM. *2019 International Conference on Robotics and Automation (ICRA)*, 2019.

[16] Chenjie Wang, Bin Luo, Yun Zhang, Qing Zhao, Lu-Jun Yin, Wei Wang, Xin Su, Yajun Wang, and Chengyuan Li. DymSLAM: 4d dynamic scene reconstruction based on geometrical motion segmentation. *IEEE Robotics and Automation Letters*, 2021.

[17] Yifei Ren, Binbin Xu, Christopher L. Choi, and Stefan Leutenegger. Visual-inertial multi-instance dynamic SLAM with object-level relocalisation. In *arXiv*, 2022.

[18] Marius Fehr, Fadri Furrer, Ivan Dryanovski, Jürgen Sturm, Igor Gilitschenski, Roland Siegwart, and Cesar Cadena. TSDF-based change detection for consistent long-term dense reconstruction and dynamic object discovery. In *2017 IEEE International Conference on Robotics and automation (ICRA)*. IEEE, 2017.

[19] Lukas Maximilian Schmid, Jeffrey A. Delmerico, Johannes L. Schönberger, Juan I. Nieto, Marc Pollefeys, Roland Siegwart, and César Cadena. Panoptic multi-TSDFs: a flexible representation for online multi-resolution volumetric mapping and long-term dynamic scene consistency. *CoRR*, 2021.

[20] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*. IEEE, 2007.

[21] Felix Endres, Jurgen Hess, Jurgen Sturm, Daniel Cremers, and Wolfram Burgard. 3-d mapping with an RGB-D camera. *Robotics, IEEE Transactions on*, 2014.

[22] Mona Gridseth and Timothy Barfoot. Towards direct localization for visual teach and repeat. In *2019 16th Conference on Computer and Robot Vision (CRV)*, 2019.

[23] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 2011.

[24] Raul Mur-Artal and Juan D Tardós. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 2017.

[25] Myriam Servières, Valérie Renaudin, Alexis Dupuis, and Nicolas Antigny. Visual and visual-inertial SLAM: State of the art, classification, and experimental benchmarking. *Journal of Sensors*, 2021.

[26] Boyu Gao, Haoxiang Lang, and Jing Ren. Stereo visual SLAM for autonomous vehicles: A review. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020.

[27] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask

r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[28] Pranav Ganti and Steven L. Waslander. Network uncertainty informed semantic feature selection for visual slam. In *2019 16th Conference on Computer and Robot Vision (CRV)*, 2019.

[29] Ioan Andrei Bârsan, Peidong Liu, Marc Pollefeys, and Andreas Geiger. Robust dense mapping for large-scale dynamic environments. *CoRR*, abs/1905.02781, 2019. URL http://arxiv.org/abs/1905.02781.

[30] Michael Strecke and Jörg Stückler. EM-fusion: Dynamic object-level SLAM with probabilistic data association. In *Proceedings IEEE/CVF International Conference on Computer Vision 2019 (ICCV)*. IEEE, 2019.

[31] Jun Zhang, Mina Henein, Robert Mahony, and Viorela Ila. VDO-SLAM: A Visual Dynamic Object-aware SLAM System. *arXiv*, 2020.

[32] Jiahui Huang, Sheng Yang, Zishuo Zhao, Yu-Kun Lai, and Shi-Min Hu. Clusterslam: A SLAM backend for simultaneous rigid body clustering and motion estimation. *Computational Visual Media*, 2021.

[33] Clara Gomez, Alejandra C. Hernandez, Erik Derner, Ramon Barber, and Robert Babuška. Object-based pose graph for dynamic indoor environments. *IEEE Robotics and Automation Letters*, 2020.

[34] Aisha Walcott-Bryant, Michael Kaess, Hordur Johannsson, and John J Leonard. Dynamic pose graph SLAM: Long-term mapping in low dynamic environments. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012.

[35] David M. Rosen, Julian Mason, and John J. Leonard. Towards lifelong feature-based mapping in semi-static environments. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1063–1070, 2016.

[36] John G. Rogers, Alexander J. B. Trevor, Carlos Nieto-Granda, and Henrik I. Christensen. SLAM with expectation maximization for moveable object tracking. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2010.

[37] Lingzhu Xiang, Zhile Ren, Mengrui Ni, and Odest Chadwicke Jenkins. Robust graph SLAM in dynamic environments with moving landmarks. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015.

[38] Sean L. Bowman, Nikolay A. Atanasov, Kostas Daniilidis, and George J. Pappas. Probabilistic data association for semantic slam. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1722–1729, 2017.

[39] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can and not as i say: Grounding language in robotic affordances. In *arXiv preprint arXiv:2204.01691*, 2022.

[40] Xin Zhou, Xiangyong Wen, Zhepei Wang, Yuman Gao, Haojia Li, Qianhao Wang, Tiankai Yang, Haojian Lu, Yanjun Cao, Chao Xu, and Fei Gao. Swarm of micro flying robots in the wild. *Science Robotics*, 2022.

[41] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[42] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. G2o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, 2011.

[43] Edwin Olson and Pratik Agarwal. Inference on networks of mixtures for robust robot mapping. In *Proceedings of Robotics: Science and Systems*, 2012.