

Khronos: A Unified Approach for Spatio-Temporal Metric-Semantic SLAM in Dynamic Environments

Lukas Schmid¹, Marcus Abate¹, Yun Chang¹, and Luca Carlone¹

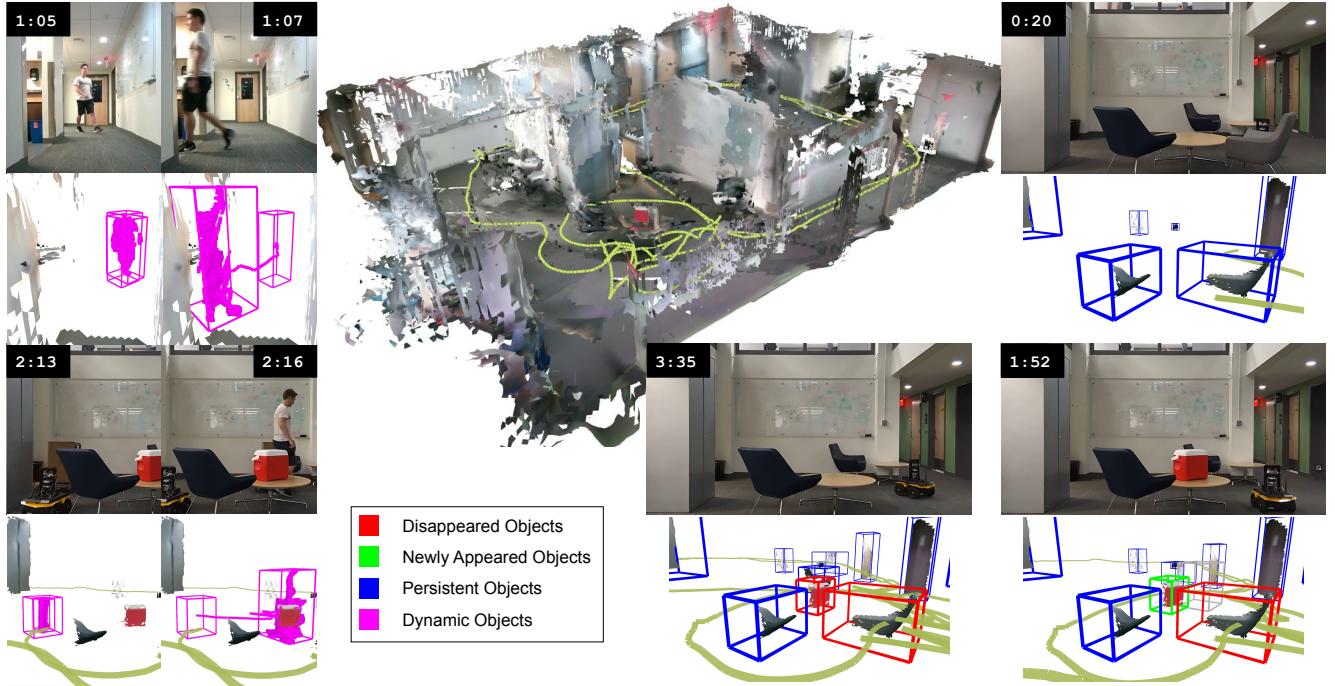


Fig. 1: We propose **Khronos**, a unified approach to reason about short-term dynamics and long-term changes when performing metric-semantic simultaneous mapping and localization (SLAM) in dynamic environments. A few instances from Khronos’ *spatio-temporal* map, representing the scene state at all times, are shown above. **Short-term dynamics** (left) are shown in magenta and compared against observed human actions over the corresponding time interval. We show the current and initial bounding box around the detected moving points as well as the centroid trajectory. Both humans and inanimate objects such as a cart (bottom left) are detected. **Long-term changes** (right) are shown for three time instances of the same scene. The earliest instance is at time 0:20 (top right). While the robot is moving through the hallways, a chair is removed and a red cooler is placed on top of the table; these changes are detected as the robot revisits and closes the loop at time 1:52 (bottom right). Lastly, the cooler is removed again, which is detected by the robot at time 3:35.

Abstract—Perceiving and understanding highly dynamic and changing environments is a crucial capability for robot autonomy. While large strides have been made towards developing dynamic SLAM approaches that estimate the robot pose accurately, a lesser emphasis has been put on the construction of dense *spatio-temporal* representations of the robot environment. A detailed understanding of the scene and its evolution through time is crucial for long-term robot autonomy and essential to tasks that require long-term reasoning, such as operating effectively in environments shared with humans and other agents and thus are subject to short and long-term dynamics. To address this challenge, this work defines the *Spatio-temporal Metric-semantic SLAM* (SMS) problem, and presents a framework to factorize and solve it efficiently. We show that the proposed factorization suggests a natural organization of a spatio-temporal perception system, where a fast process tracks short-term dynamics in an active temporal window,

while a slower process reasons over long-term changes in the environment using a factor graph formulation. We provide an efficient implementation of the proposed spatio-temporal perception approach, that we call *Khronos*, and show that it unifies existing interpretations of short-term and long-term dynamics and is able to construct a dense spatio-temporal map in real-time. We provide simulated and real results, showing that the spatio-temporal maps built by Khronos are an accurate reflection of a 3D scene over time and that Khronos outperforms baselines across multiple metrics. We further validate our approach on two heterogeneous robots in challenging, large-scale real-world environments.

I. INTRODUCTION

In order to operate safely and effectively in human-populated environments, a robot needs to have a sufficient understanding of the world around it. Such shared spaces are often highly dynamic, with people, robots, and other entities constantly moving, interacting, and modifying the scene. For a robot to operate in such circumstances, it is not sufficient to build a world model just for a single snapshot in time. Instead, the robot should be also able to reason over the state

¹MIT SPARK Lab, Massachusetts Institute of Technology, MA, USA.
 {lschmid, mabate, yunchang, lcarlone}@mit.edu

This work is partially funded by the Amazon Science Hub “Next-Generation Spatial AI for Human-Centric Robotics” project, the ARL DCIST program, the ONR RAPID program, and the Swiss National Science Foundation (SNSF) grant No. 214489.

of the scene at past times, inferring how the scene might have changed across multiple observations. Such capabilities are essential for a variety of applications that require reasoning over longer time spans, ranging from household and service robotics, to industrial construction or work-site monitoring, where robots are not only required to operate in highly dynamic environments, but also to keep track of — or reason about — the evolution of the environment from the past to more intelligently carry-out tasks efficiently.

Metric-semantic simultaneous localization and mapping (SLAM) [38] allows a robot to construct a semantically annotated geometric representation of a scene in real-time. Geometric information is critical for robots to navigate safely and to manipulate objects, while semantic information provides the understanding for a robot to execute human instructions and to provide humans with models of the environment that are easy to understand. In order to build these dense metric-semantic representations in real-time, it is common to assume that the world is static and focus on robustly fusing noisy geometric and semantic measurements into a metric-semantic model [28, 21]. Even though this is a valid assumption for certain robotic applications, it limits the generality of the types of environments a robot can operate in, along with the tasks it can be assigned to carry out. On the other hand, there exists an extensive body of work addressing SLAM in dynamic environments [35, 8, 5, 55, 47, 56]. These approaches show impressive capabilities in improving robot localization and state estimation in spite of moving entities within view of the robot. However, these prior works mostly focus on *short-term* dynamics, such as people or objects currently moving in front of the camera, and the corresponding literature has often been disconnected from the body of work focusing on *long-term* change detection [22, 32, 42, 34], where the scene undergoes substantial changes (e.g., furniture being rearranged) while the robot is not directly observing it. Real environments undergo both short-term and long-term changes and the literature currently lacks a unifying approach that can reason over both. Moreover, to enable robots to effectively work alongside other humans and robots, such a framework needs to build this understanding of the world online during robot operation with the limited information and computational resources available.

To this end, we introduce the *Spatio-temporal Metric-semantic SLAM* (SMS) problem, which aims at building a dense metric-semantic model of the world at all times incrementally as the robot navigates the scene. We present a unified framework to tackle the SMS problem. The central idea of our approach is to develop a new factorization of the SMS problem based on spatio-temporal local consistency, which allows for the disentanglement of errors arising from sensing noise, state estimation errors, dynamic objects, and long-term changes in the scene. We integrate this insight into a spatio-temporal perception system, named *Khronos*, which is the first real-time metric-semantic system capable of building a spatio-temporal map of the scene. We thoroughly evaluate our method in several simulated scenes with detailed annotations on background reconstruction, object

detection, motion tracking, and change detection and on multiple robotic platforms navigating highly dynamic real-world environments. We make the following contributions:

- We formalize the Spatio-temporal Metric-semantic SLAM (SMS) problem, which allows a robot to build a dense metric-semantic understanding of the surrounding environment and its evolution over time.
- We propose a novel factorization of the SMS problem, which provides a unifying lens for existing interpretations focusing on short-term and long-term dynamics.
- We present Khronos, the first spatio-temporal metric-semantic perception system, composed of novel algorithms for asynchronous local mapping and deformable global change detection.

We release our implementation and datasets open-source.¹

II. RELATED WORKS

Metric-semantic SLAM. The goal of metric-semantic SLAM is to build a semantically annotated 3D map during online robot operation. Prominently, voxel-based methods [48, 23, 44] can incrementally fuse noisy detections into the map. However, their rigid grid-structure makes them susceptible to state estimation drift. Alternatively, object-level SLAM methods [28, 21] and the related landmark-based methods [4, 30] refine the state estimate and maintain semantic information about objects in the scene. To achieve global consistency of the dense map, other representations such as surfels [7], meshes [45], or submap-based methods [57, 6] have been proposed. Recently, also neural representations including NeRF [26, 40] and Gaussian-Splatting-based mapping methods [19, 27, 53] have been proposed. However, these methods oftentimes require powerful computers, making them unsuitable for real-time robot operation. To jointly optimize for the robot poses and semantic reconstruction of the scene, the methods above generally make the simplifying assumption that the world is static.

Dynamic SLAM. To address this limitation, a large body of work has emerged, which can be grouped into two main categories. First, sparse SLAM methods focus on improving state estimation performance [35, 8, 5, 55, 47, 56, 1] — typically by removing dynamic objects from the SLAM problem [55, 47, 56]. Alternatively, dynamic entities such as cars [2, 14] or humans [15] can be integrated into the estimation problem, leading to improved state estimates if good motion priors for these entities are available. Second, simultaneous tracking and reconstruction approaches [41, 24, 49, 51, 13, 52] are able to generate dense models of single [41, 24] or multiple [49, 51, 13, 52], rigid objects moving in front of the camera. However, these approaches are often limited to table-top scenes. Dynablox [43] and similar approaches [37, 29] can simultaneously perform motion detection and dense background reconstruction in larger scenes. Nonetheless, these methods often rely on tracking of incremental motion against a static background and may not generalize well to long-term changes in the scene.

¹Released upon acceptance at <https://github.com/MIT-SPARK/Khronos>.

Change Detection. Identifying long-term changes is traditionally addressed in a multi-session scenario, where two observations of the static scene before and after changes occurred are compared [22, 32, 58, 50]. Notably, Bore *et al.* [3] detect and also track general long-term dynamic objects using a filtered probabilistic motion model. Recently, the first approaches [42] for online long-term consistent mapping have emerged. Panoptic Mapping [42] leverages semantic consistency of foreground objects and background classes to maintain a volumetric map in changing scenes. Fu *et al.* extend this idea with local registration and volumetric object descriptors [11], as well as neural object models [10]. Recently, POCD [33] and POV-SLAM [34] approach the change-detection problem in a SLAM scheme and propose an object-aware SLAM pipeline to track and reconstruct object-level long-term changes in a factor-graph formulation.

Spatio-temporal Mapping. To the best of our knowledge, there are comparatively fewer works that combine short-term dynamics and long-term dynamics. Recently, Soares *et al.* [46] showcase Changing-SLAM, which increases state-estimation robustness by accounting for both short-term and long-term dynamics in the scene. They build a map of sparse key points and employ a Bayesian filtering approach to detect short-term dynamics and data association to detect long-term changes. Any detected dynamic points are then removed from the SLAM problem. While this accounts for the effect of both short and long-term dynamics on state estimation, they do not focus on building a rich representation of the scene and an understanding of its evolution. In contrast, to our knowledge, our proposed method is the first to generate a spatio-temporal metric-semantic map in real-time, jointly optimizing for the robot poses and an explicit dense semantic representation of the scene at all times.

III. PROBLEM STATEMENT

To define the spatio-temporal metric-semantic SLAM (SMS) problem, we consider a scene that is composed of objects $O_i \in \mathcal{O}$. The entire background of the scene is also represented as a static object $O_{BG} \in \mathcal{O}$. We denote the state of each object i at time t as O_i^t :

$$O_i^t = \{\Omega_i^t, T_{WO_i}^t, L_i\}, \quad (1)$$

where Ω_i^t denotes its surface, $T_{WO_i}^t$ its pose w.r.t. the world frame W , and L_i the semantic label of O_i , at discrete time $t = 0, 1, \dots, T$. The background object O_{BG} is constant for all times t . We denote the robot poses X as:

$$X = \{X^t\}_{t=0}^{t=T}, \quad X^t = T_{WR}^t. \quad (2)$$

In the following, we use the shorthand notation for indexed variables of omitting the index and/or time step to refer to all existing indices and/or time steps, e.g.:

$$O_i = \{O_i^t\}_{t=0}^{t=T}, \quad O = \{O_i^t\}_{i=0}^{i=N_t} \quad (3)$$

At each time t , the robot observes a volume V^t which gives rise to measurements Z_j^t :

$$Z_j^t = \{\Omega_j^t, T_{RZ_j}^t, L_j^t\}, \quad (4)$$

$$Z_j^t \sim O_i^t \oplus \eta_O \text{ if } O_i^t \in V^t. \quad (5)$$

where Ω_j^t is a surface measurement (i.e. from an RGBD camera), $T_{RZ_j}^t$ is the pose of the object surface w.r.t. the robot frame R at time t , and L_j^t the observed semantic label of the surface at time t . We summarize all measurement noise in the operation $\oplus \eta_O$, which includes errors such as surface measurement inaccuracies, but also missed or hallucinated measurements, or noisy semantic information. Besides taking visual observations of the volume V^t , the robot makes an odometry measurement:

$$\Phi^t \sim (X^{t-1})^{-1} X^t \boxplus \eta_\Phi, \quad (6)$$

where \boxplus denotes addition on $SE(3)$. The goal of the SMS problem is to build a spatio-temporal understanding of the scene in real time, i.e., at each current time T , our goal is to estimate the state of the scene at all previous times $t \leq T$. This can be posed as a maximum-a-posteriori (MAP) estimation problem for each T :

$$O^*, X^* = \arg \max_{O, X} \mathbb{P}(O, X | Z, \Phi). \quad (7)$$

IV. FRAGMENTS AND FACTORIZATION

The MAP estimate in (7) is hard to compute for a number of reasons. First, since the number of objects as well as their attributes can change, there is a large number of unknown variables and comparably fewer measurements. Second, the coupling between sensing noise, imperfect state estimation, and moving and changing scenes introduces a high degree of interdependence of all variables. This leads to having to keep all observations in memory, which causes the problem to scale poorly with space and time. Intuitively, this reflects the challenge that disagreements between the current measurement and map belief can originate from noise in the sensing process, erroneous state estimates, objects currently moving, or the scene having changed over time.

To overcome these challenges, we propose a novel factorization of the above problem based on the key idea of *spatio-temporal local consistency*. Formally, we assume that:

$$\left\| \left(\prod_{t'=-\tau}^t \Phi^{t'} \right) \boxminus (T_{WR}^{t-\tau})^{-1} T_{WR}^t \right\| < \epsilon_s \quad \forall t, \forall \tau < \delta, \quad (8)$$

$$\|O_i^t \ominus O_i^{t-\tau}\| < \epsilon_t \quad \forall t, \forall \tau < \delta. \quad (9)$$

Intuitively, this means that both errors in state estimation (8) and changes in the scene (9), while they may grow large over time, are small for short time intervals δ .

The central idea of our approach is to decouple the dependencies in problem (7) through the introduction of a set of latent variables Y , which we will refer to as “object fragments”, that serve as intermediaries between Z and O . Each object fragment can be thought of as a partial view of an object constructed by collecting multiple locally consistent surface measurements over a short time interval (a *fragment* of time). Importantly, we define Y as the minimal partitioning of the observations Z of O such that local consistency (8), (9) holds within the observations each Y_k . Practically, this can be thought of as breaking each O_i into sequences of

timestamps, the fragments, where the time between consecutive observations Z is $< \delta$, as illustrated in Fig. 2 (left). We define properties of Y_k^t in the robot frame R at time t as:

$$Y_k = \{\Omega_k, T_{RY_k}, L_k\}. \quad (10)$$

To summarize, observations Z are accumulated into object fragments Y based on local consistency, and each true object O is a collection of fragments Y . The challenge, however, is that all observations Z are in robot frame R , whereas the goal is to estimate O in W , and this introduces a strong global coupling between Z , X and O , as X relates the W and R frames. We note that the fragments Y fully specify the objects O , hence we can express O as a function of Y :

$$\mathbb{P}(O, X, Y | Z, \Phi) = \mathbb{P}(O, X | Y, \Phi) \mathbb{P}(Y | Z, \Phi). \quad (11)$$

Furthermore, since each $Y_k \in Y$ only depends on non-overlapping sets of measurements, we can assume that all $Y_k \in Y$ are conditionally independent, such that

$$\mathbb{P}(O, X, Y | Z, \Phi) = \mathbb{P}(O, X | Y, \Phi) \prod_k \mathbb{P}(Y_k | Z, \Phi). \quad (12)$$

We note that each Y_k only depends on a subset of measurements $\bar{Z}_k \subseteq Z$, $\bar{\Phi}_k \subseteq \Phi$ that fall into its temporal window, thus yielding a first key decomposition of the SMS problem:

$$\mathbb{P}(O, X, Y | Z, \Phi) = \underbrace{\mathbb{P}(O, X | Y, \Phi)}_{\text{Global estimation}} \prod_k \underbrace{\mathbb{P}(Y_k | \bar{Z}_k, \bar{\Phi}_k)}_{\text{Local estimation}}. \quad (13)$$

This formulation isolates the effect of sensing noise in $\mathbb{P}(Y_k | \bar{Z}_k, \bar{\Phi}_k)$, the likelihood of a fragment $Y_k \in Y$ given measurements in the corresponding temporal window. However, the global part of (13) still contains a coupling between spatial and temporal sources of inconsistency that makes this a hard estimation problem. To alleviate this, we again leverage our definition of Y_k as partial observations of an object O_i within a time fragment. Since Y fully specify O , if their poses $T_{WY_k} = X \cdot T_{RY_k}$ are known we can rewrite the global estimation part of (13) as:

$$\mathbb{P}(O, X | Y, \Phi) = \mathbb{P}(O | X, Y) \mathbb{P}(X | Y, \Phi). \quad (14)$$

Since for a single object O_i , all relevant information is captured in their respective segments $\bar{Y}_i \subseteq Y$, if the associations $A_i : Y_k \mapsto \bar{Y}_i$ of fragments Y_k to objects O_i are known, this further simplifies to (15), as illustrated in Fig. 2 (right):

$$\mathbb{P}(O, X, A | Y, \Phi) = \prod_i \mathbb{P}(O_i | \bar{Y}_i, X) \mathbb{P}(X, A | Y, \Phi). \quad (15)$$

The global estimation problem is thus decomposed into two sub-problems: the fragment reconciliation term, from which the objects are inferred from their fragments, and the second term, which takes the form of a landmark-based SLAM setup. Finally, substituting (15) into (13) yields:

$$\begin{aligned} &\mathbb{P}(O, X, Y, A | Z, \Phi) = \\ &\underbrace{\prod_i \mathbb{P}(O_i | \bar{Y}_i, X)}_{\text{Fragment reconciliation}} \underbrace{\mathbb{P}(X, A | Y, \Phi)}_{\text{SLAM}} \prod_k \underbrace{\mathbb{P}(Y_k | \bar{Z}_k, \bar{\Phi}_k)}_{\text{Local estimation}}. \end{aligned} \quad (16)$$

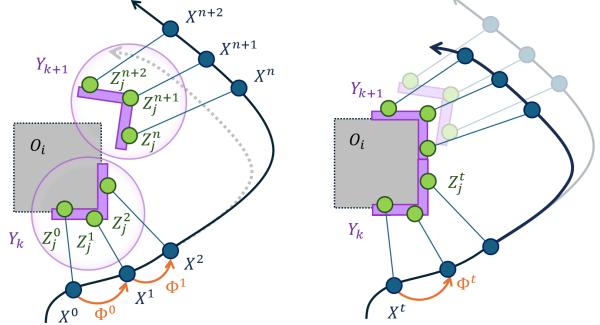


Fig. 2: Key variables in our formulation. All observations Z_j^t of O_i are grouped into fragments Y_k , s.t. local consistency holds. Once it breaks, i.e., as $t(n) > t(2) + \delta$, measurements are grouped into a new fragment Y_{k+1} . This allows estimating Y_k from the closed sets $\bar{Z}_k = \{Z_j^0, Z_j^1, Z_j^2\}$ and $\bar{\Phi}_k = \{\Phi^0, \Phi^1\}$ (left). Once the robot and fragment poses as well as associations are optimized, the object O_i is fully specified by all of its fragments $\bar{Y}_i = \{Y_k, Y_{k+1}\}$ (right).

Thus, with only minimal assumptions, we have obtained a well structured problem (16) that we optimize as a surrogate for (7). More importantly, this approach provides a unified framework that naturally gives rise to existing interpretations. First, short-term and long-term dynamics as defined in [25] naturally emerge, where *all* short-term dynamics, characterized by observations of continuous motion, are captured in the local part of (16), and *all* long-term dynamics, characterized by observations of abrupt changes, are captured in the global part of (16). Second, the difference between modeling objects based on observations and inferring what happened while not observed is clearly represented by considering the O_i^t for which $\exists Y_k^t$, as Y_k group all observations of an object O_i , and $\nexists Y_k^t$, respectively. Third, this formulation naturally enforces semantic consistency as introduced in [42]. Finally, the resulting problem structure has important algorithmic properties, further detailed in Sec. V.

V. KHRONOS

This section introduces *Khronos*, our system to optimize (16). We follow (16) and split the problem into three components, an *active window*, a *global optimization*, and a *reconciliation* component, as shown in Fig. 3.

A. Local Estimation via an Active Window

We refer to the local estimation component as the *active window*. Its goal is to solve the local term in (16) by incrementally estimating a set of fragments Y_k from observations Z, Φ such that local consistency (8), (9) is satisfied.

Reconstruction. We first reconstruct the static background. We point out that, while the formulation holds for any surface representation Ω , we implement Khronos using meshes to model static surfaces Ω_{BG} . To this end, a volumetric map is incrementally allocated around the robot and projective TSDF fusion is performed to estimate Ω_{BG} . We then obtain candidate observations Z in every frame from raw RGBD data. Following the definition of (1), different cues can be used to extract Z from sensor data. First, we leverage semantic masks provided in the input frame. Second, we employ geometric motion detection to separate objects from

the background. We therefore augment the volumetric local map following the approach of [43], using the motion cue that points falling into previously observed free space must be dynamic. However, since we only consider a short temporal window δ as introduced in (8), the *Occupancy*, *Sensor Sparsity* and *State Estimation* terms of [43] are dropped. In summary, we allocate one measurement Z_j^t for every mask, where Ω_j^t is the set of 3D points in that mask, T_{RZ_j} is the transform to the sensor, and L_z is an integer or feature vector for closed or open-set semantics, respectively (see Sec. VI-C). **Tracking.** To estimate fragments Y that best explain the observations Z , we generate a pool of object hypotheses \hat{Y} . Since changes within the active window are small (9), we can greedily associate new observations Z_j^t to the best fitting hypothesis \hat{Y}_k by computing the volumetric IoU between each Z_j^t and \hat{Y}_k . The IoU can efficiently be computed using a grid-aligned voxel-filter on the points Ω_k and Ω_j . We associate semantic Z_j^t to \hat{Y}_k of matching labels L_k , dynamic Z_j^t to the closest dynamic \hat{Y}_k , and allow cross-associations representing semantic-dynamic observations. For each Z_j^t that was not associated, a new hypothesis \hat{Y}_k is added to the pool. Following (13), once local consistency is broken, no future observations can be part of the same Y , and these \hat{Y} can be removed from the pool. We then estimate the probability of that hypothesis \hat{Y} representing a true fragment Y representing an object of class L_k by rejecting \hat{Y}_k that have less than $\tau_Z = 15$ observations and dynamic \hat{Y}_k that have moved less than $\tau_D = 1\text{ m}$. We then marginalize all observations \bar{Z}_k belonging to that Y_k . It is worth to point out that this gives an enormous freedom in choosing suitable representations depending on the task and computation power at hand, given that all relevant observations \bar{Z}_k are known at extraction time. In this approach, we choose to use TSDF fusion to reconstruct the surfaces of static objects as meshes with adaptive resolutions Ω_k , and use sequences of pointclouds to represent deformable and dynamic object surfaces Ω_k . Similarly, we track local consistency of the background and extract vertices that exit the active window. This asynchronous tracking of every entity in the active window guarantees that local consistency holds and naturally handles partial or erroneous observations. Simultaneously, it has the large advantage that object properties can be estimated when all data is available.

B. Global Optimization

The global optimization module addresses the second term in (16); the local estimates from the active window are optimized and updated for reconciliation. Globally, we jointly estimate the poses of the robot X , the poses of the fragments Y , and the dense background mesh Ω_{BG} . In particular, we construct a deformation graph as described in [16] and augment it with fragments Y_k . Specifically, the nodes of the deformation graph correspond to robot poses X and mesh control points P_M selected from Ω_{BG} as in [16], and a new set of fragment poses T_{WY_k} . We initialize T_{WY_k} at the centroid of Ω_k with unit orientation, representing the reconstruction frame of Ω_k . We connect T_{WY_k} to the

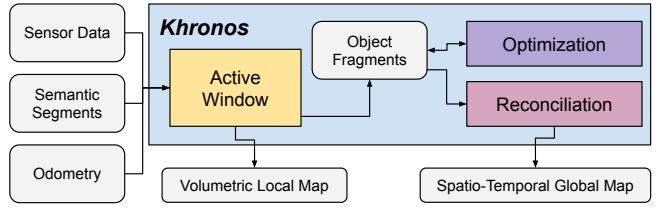


Fig. 3: Khronos takes in robot odometry Φ , and semantic and RGBD inputs Z . The active window (Sec. V-A) uses these to estimate a local representation and object fragments Y . We then perform global optimization (Sec. V-B) to estimate robot poses X and fragment association A . Finally, the optimized fragment states are reconciled (Sec. V-C) to estimate the spatio-temporal map of the scene.

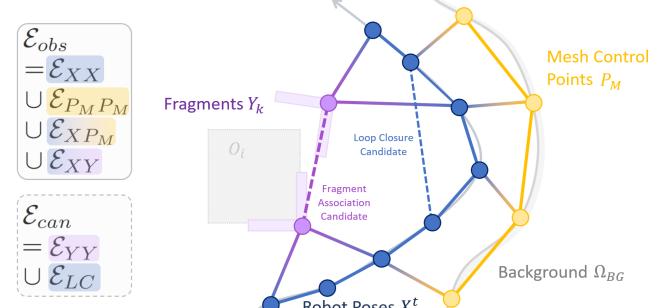


Fig. 4: Overview of the constructed deformation graph \mathcal{E} .

robot pose graph with edges \mathcal{E}_{XY} . Each T_{WY_k} is always connected to the robot poses X_f and X_l , corresponding to when the fragment was *first* and *last* observed, respectively. This reflects the robot-centric measurement of Y_k (10) and ensures that when the robot poses are optimized, the fragment poses are updated correspondingly.

To compute associations A between fragments Y when the robot re-observes an object, we generate candidate associations as edges \mathcal{E}_{YY} between fragments that have identical labels L_k (or similar L_k for open-set semantics), and whose bounding boxes overlap. For this initial work, we model \mathcal{E}_{YY} as a pure translation constraint between T_{WY_a} and T_{WY_b} , reflecting that the centroids of Ω_a and Ω_b should be close if they stem from the same object. We note that full 6DOF registration between Ω_a and Ω_b can naturally be integrated into our framework, but leave this for future work. Finally, we add candidate edges \mathcal{E}_{LC} for loop closures provided by the odometry input.

Hence, the edges of the final deformation graph consist of *observed* edges $\mathcal{E}_{obs} = \mathcal{E}_{XX} \cup \mathcal{E}_{PMPM} \cup \mathcal{E}_{XP_M} \cup \mathcal{E}_{XY}$, and *candidate* edges $\mathcal{E}_{can} = \mathcal{E}_{YY} \cup \mathcal{E}_{LC}$, thus $\mathcal{E} = \mathcal{E}_{obs} \cup \mathcal{E}_{can}$. An overview of this is shown in Fig. 4. The estimates of the robot poses, fragment positions, and the background mesh is then obtained by finding a solution $\mathcal{T} = X \cup P_M \cup T_{WY}$ of the robust pose graph optimization problem:

$$\begin{aligned} \mathcal{T}^* = \arg \min_{\mathbf{T}_1, \dots, \mathbf{T}_n \in \mathcal{T}} & \left[\sum_{\substack{(i,j) \in \mathcal{E}_{obs} \\ \omega_{ij} \in \{0,1\}}} \|\mathbf{T}_i^{-1} \mathbf{T}_j \boxminus \bar{\mathbf{T}}_{ij}\|_{\Lambda_{ij}}^2 \right. \\ & \left. + \sum_{(i,j) \in \mathcal{E}_{can}} \left(\omega_{ij} \|\mathbf{T}_i^{-1} \mathbf{T}_j \boxminus \bar{\mathbf{T}}_{ij}\|_{\Lambda_{ij}}^2 + (1 - \omega_{ij}) \bar{c}^2 \right) \right], \end{aligned} \quad (17)$$

where $\mathbf{T}_i \in SE(3)$ and $\mathbf{T}_j \in SE(3)$ are pairs of 3D poses in \mathcal{T} , $\bar{\mathbf{T}}_{ij} \in SE(3)$ is the relative measurement

associated to each edge $(i, j) \in \mathcal{E}$, and \square represents a subtraction operation on $SE(3)$ transforms. We use the notation $\|\mathbf{M}\|_{\Lambda}^2 = \text{tr}(\mathbf{M}\Lambda\mathbf{M}^T)$ and the 4×4 positive semi-definite matrix Λ_{ij} is chosen as the inverse of the covariances for the edges. For the edges in \mathcal{E}_{YY} , we set Λ_{ij} to $\text{diag}([0 \ 0 \ 0 \ \lambda])$, where the zeros cancel out the rotation residual and $\lambda = 1$ is chosen permissively since we use the centroids of the Ω_k as reference for \mathcal{T}_{Y_k} .

To account for erroneous measurements and perceptual aliasing, we solve the pose graph optimization problem with the Truncated Least Squares (TLS) loss, where we optimize also for binary variables ω_{ij} such that $\omega_{ij} = 1$ for correct inlier measurements and $\omega_{ij} = 0$ for outliers, where \bar{c} is the outlier truncation cost. Using the binary weights, we can classify the correctness of the candidate associations $\mathcal{E}_{YY} \cup \mathcal{E}_{LC}$. In practice, we solve the TLS problem with Graduated Non-Convexity (GNC) [54] in GTSAM [9]. From the solution \mathcal{T}^* , we directly update the robot and object fragment poses, collect the correct associations ω_{ij} , and update the dense background mesh using the mesh control points P_M^* as described in [39].

C. Reconciliation

Finally, the goal of reconciliation is to estimate the state of the scene at all times $t < T$, given the optimized fragments Y at time T . It is important to note that fragments only contain positive observations, i.e., information about the presence of detected objects, but not about their absence. To resolve this evidence-of-absence vs. absence-of-evidence problem, we perform an additional geometric verification step.

Deformable Change Detection. We observe that, while no volumetric information is stored in our surface representation, this information is partially implicitly captured in the background and robot poses. The central idea is that rays connecting a background vertex and a corresponding robot pose that observed it were in free space at that time. However, since the robot and background poses can continually change during global optimization, integrating this information into a free-space map is not tractable during online operation. Instead, we approximate this global free-space information by a *library of rays*. Whenever a background vertex is extracted from the active window, we create a representative ray of that vertex $\mathbf{p}_v \in \mathbb{R}^3$ to the robot position \mathbf{p}_r in the middle of its observation window. For efficient lookup, this is implemented by storing the indices of the vertex and view point in a coarse global hash map. These can efficiently and incrementally be added, and allow for the robot and background points to move freely within their grid cells. When large position changes are detected, e.g. in the case of a loop closure, the hash map is re-computed from scratch. To detect evidence of absence or presence, we query points \mathbf{p}_q on the surface of fragments Y_k in this library of rays. For each candidate ray returned by the hash map, we compute the distance d_r of the query point to the ray to determine whether the point is on the ray, and the depth distance d_d along the ray:

$$d_r = \|(\mathbf{p}_q - \mathbf{p}_r) \times (\mathbf{p}_r - \mathbf{p}_v)\| / \|\mathbf{p}_q - \mathbf{p}_r\|, \quad (18)$$

$$d_d = (\mathbf{p}_q - \mathbf{p}_r) \cdot (\mathbf{p}_v - \mathbf{p}_r) / \|\mathbf{p}_q - \mathbf{p}_r\|. \quad (19)$$

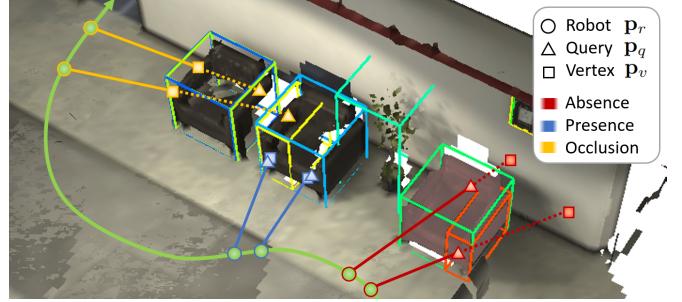


Fig. 5: Deformable change detection. A \mathbf{p}_q in front of \mathbf{p}_v indicates the ray would pass through \mathbf{p}_q , and \mathbf{p}_q is thus absent. Similarly, \mathbf{p}_q near or behind \mathbf{p}_v indicates presence or an occlusion, respectively. All \mathbf{p}_r and \mathbf{p}_v can freely move, rendering the approach deformable.

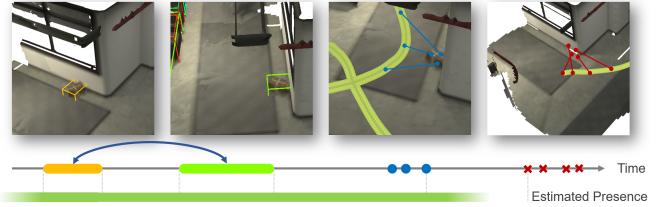


Fig. 6: Reconciliation. An object is visited twice, resulting in two fragments (orange and green). Fragments are associated via global optimization (blue arrow). Even without semantic detection, geometric verification provides evidence of presence or absence. We estimate the object must have disappeared in between the last presence and first absence observation.

Distances d_d longer than that of the vertex indicate the point was occluded, distances similar (within $d_{ray} = 30$ cm) to the vertex indicate geometric consistency with that observation, and short distances indicate evidence of absence. An example of this logic is shown in Fig. 5. In this way, we obtain all timestamps of rays reporting evidence of absence or presence of an object. Since the robot poses and mesh are continuously optimized, a timestamp is only considered reliable evidence of absence if at least $c_{ray} = 60\%$ percent of rays within a temporal window of $\tau_{ray} = 5$ s mark the object as absent.

Reconciliation. Finally, we can estimate the times each object was present in between fragments Y . To this end, we compute the latest absence of evidence before the fragment Y was first observed and the earliest after it exited the active window. Similarly, we compute the earliest and latest evidence of presence within this window above. Intuitively, this reflects the logic that an object must have newly appeared some time between when its location was last observed to be empty and when the object was first observed in that location, and inversely for its disappearance. Assuming that objects have a uniform probability of appearing or disappearing, the minimum expected error estimate can easily be shown to be the middle of that window. An example is shown in Fig. 6.

VI. EXPERIMENTS

A. Experimental Setup

Datasets. To thoroughly evaluate the proposed method, a dataset with both short and long-term dynamics in a single sequence and detailed spatio-temporal annotations is required. As, to the best of our knowledge, no dataset with all these

TABLE I: 4D background reconstruction, static object, dynamic object, and change detection metrics [%]. Higher is better for all metrics.

Scene	Poses	Method	Background			Objects			Dynamics			Changes		
			Pre	Rec	F1									
Apartment	GT	Hydra [16]	94.9	83.1	87.7	91.0	28.0	42.3	-	-	-	-	-	-
		Dynablox [43]	84.6	89.1	86.2	-	-	-	48.9	95.0	61.3	-	-	-
		Panoptic Mapping [42]	95.9	56.1	70.3	94.6	49.7	64.3	-	-	-	38.2	56.1	56.1
		Khronos (ours)	96.8	87.6	91.2	91.4	83.9	75.3	90.4	78.6	84.1	31.3	69.1	64.6
	Kimera [39]	Hydra [16]	94.3	81.5	86.5	93.7	32.3	47.0	-	-	-	-	-	-
		Dynablox [43]	83.5	88.3	85.2	-	-	-	44.4	95.1	56.2	-	-	-
		Panoptic Mapping [42]	95.2	52.2	67.1	95.6	46.3	61.8	-	-	-	23.4	65.0	47.4
		Khronos (ours)	95.4	86.7	90.1	90.7	57.1	69.2	96.4	78.5	86.3	21.3	62.9	45.6
Office	GT	Hydra [16]	96.8	69.4	78.0	95.6	59.1	68.8	-	-	-	-	-	-
		Dynablox [43]	85.9	73.2	76.2	-	-	-	22.6	47.2	23.0	-	-	-
		Panoptic Mapping [42]	97.5	70.8	79.0	96.3	58.0	68.7	-	-	-	33.1	32.9	60.4
		Khronos (ours)	96.3	73.7	80.6	98.7	63.9	74.2	96.0	59.7	73.2	24.5	54.2	66.2
	Kimera [39]	Hydra [16]	83.7	56.8	65.2	96.9	49.6	62.1	-	-	-	-	-	-
		Dynablox [43]	70.8	56.5	60.3	-	-	-	23.2	43.1	23.5	-	-	-
		Panoptic Mapping [42]	82.6	56.9	64.8	95.9	54.9	66.3	-	-	-	9.6	33.5	38.6
		Khronos (ours)	83.9	60.3	67.6	98.3	62.4	73.1	59.3	54.7	53.9	25.8	52.2	62.0

features exists yet, we create two scenes using the photo-realistic simulator TESSE [39]. A smaller *Apartment* dataset consists of several visits of a residential scene. The sequence is 87s long, with a robot speed of up to 1m/s over a trajectory of 39m, 64 static objects, 10 dynamic objects, and 6 long-term object changes. A second large-scale *Office* dataset explores an extensive office scene, turning a small loop in the middle, and finally closing a large loop when revisiting some of the rooms near the start pose. The sequence is 217s long, with a robot speed of up to 1m/s over a trajectory of 181m, 196 objects, 6 dynamic objects, and 8 long-term object changes. The dynamic objects include people and a football bouncing around. The changing objects range from large furniture to small household items (e.g. desks, chairs, boxes, vases). We further validate our approach in two different real-world environments on two heterogeneous robots: a Clearpath Jackal equipped with an Intel Realsense D455 RGBD camera and a Boston Dynamics Spot. We use OneFormer [17], a transformer-based semantic segmentation network, to produce the semantic segments for Khronos.

Metrics. We evaluate the ability of Khronos to reconstruct the *background*, model static *objects*, capture short-term *dynamics*, and detect long-term *changes*. For each category, we compute the metrics $\mathcal{L} = \{\text{precision}, \text{recall}, \text{F1-score}\}$. For the background, we consider each vertex that has a corresponding ground truth (GT) vertex within 20cm a positive. For static and dynamic objects, we consider each object that has a corresponding GT object a positive. For changes, we consider each object that has the same change label (newly appeared, disappeared) as the corresponding GT object a positive. It is important to point out that the recall is measured with respect to all entities existing at the evaluated time t , and not only entities already observed by the robot. To evaluate a spatio-temporal map $M_T(t)$, we compute the 4D-metric over

all robot times T and belief times t up to the final time \mathcal{T} :

$$\mathcal{L}_{4D} = \frac{1}{2\mathcal{T}^2} \int_0^{\mathcal{T}} \int_0^T \mathcal{L}(M_T(t)) dt dT. \quad (20)$$

Baselines. Since, to the best of our knowledge, there do not yet exist approaches for real-time dense metric-semantic spatio-temporal SLAM, we compare Khronos to several recent approaches specializing on the individual components. We compare against Hydra [16], a dense, globally-consistent metric-semantic perception pipeline, Dynablox [43], a simultaneous dense mapping and dynamic object detection approach, and Panoptic Mapping [42], a method for online long-term consistent volumetric mapping.

Hardware. All computation is performed on an Intel i7-12700H laptop CPU with 32GB of RAM, allowing online deployment on autonomous mobile robots.

B. Spatio-temporal Metric-semantic SLAM

Tab. I presents detailed quantitative results for Khronos and the established baselines across both simulated datasets. For fair comparison, all methods use identical resolutions $\nu = 8 \text{ cm}$ and sensing range of $r = 5 \text{ m}$. To allow accurate evaluation of the systemic components, ground truth semantics from the simulator are employed (we discuss the importance of semantics in Sec. VI-C). Finally, to assess the importance of spatial-consistency, we run each method with ground truth (GT) poses and visual-inertial-odometry estimates obtained from Kimera [39].

Background Reconstruction. We observe that all methods generally achieve comparable results in terms of background reconstruction. A slight exception is Dynablox, which does not extract static object from the background, resulting in higher recall at a lower precision. More notably, Panoptic Mapping falsely detects previous segments of background

as changed and removes them when revisiting them in the apartment scene, leading to low recall. The importance of spatial optimization becomes apparent when looking at the office scene with drift, where Khronos and Hydra are the best performing methods. The difference is not as strongly reflected in the numbers, as all methods only know the drifting odometry estimate for a long time before Khronos and Hydra can close the loop toward the end of the sequence.

Object Detection. As expected, the precision of detected objects is high given the true semantic segmentation as input. However, it is not 100% as all approaches merge and filter the input detections. However, a notable difference is apparent in recall. Since Hydra segments objects from the background, it can not capture small objects in the scene, such as the many fine items in the apartment. On the other, the two multi-resolution approaches Khronos and Panoptic Mapping achieve significantly higher recall in this scene, where the multi-hypothesis tracking and reconstruction approach of Khronos shows superior performance.

Dynamic Objects. Dynablock shows to be highly sensitive and achieves strong recall in the apartment scene. However, it also detects a number of false positives in this complex environment. On the other, Khronos is able to diminish the effect of noisy measurements through its model of persistent dynamic objects. It is apparent that the office poses a more challenging scene, where both Khronos and Dynablock detect more false positives. The reduced recall can be explained in part by the nature of some of the observed motions, such as humans walking closely in front and away from the robot, thus preventing the free-space map from being built and the motion to be detected. This is a fundamental limitation of the employed motion detection cue.

Change Detection. Observing the change detection performance, we notice that online detection of changing objects in cluttered and highly dynamic scenes is a challenging problem. Nonetheless, both methods are able to detect a significant number of changes, among some false positives. In the small apartment scene, not many meaningful loop closures are detected, and both methods show reduced performance when operating with drifting state estimates. However, the importance of spatial consistency for change detection becomes apparent in the large-scale office scene. While the precision of Panoptic Mapping breaks down for imperfect state estimates in longer sequences, Khronos is able to maintain its high performance through our joint spatio-temporal optimization and deformable change detection approach. This demonstrates that Khronos is able to build spatio-temporal maps also of large-scale environments during online robot operation.

C. Semantic Segmentation Input

Khronos is able to interface with different segmentation frontends, and the construction of the spatio-temporal map is agnostic to the input semantic or instance segmentation method. To demonstrate this, Tab. II compares the Khronos with ground-truth semantic segmentation as the input against an open-set segmentation method that clusters

TABLE II: Change detection performance with ground truth segmentation and open-set segmentation on the Apartment sequence.

Poses	GT			Kimera			
	Semantics	Pre	Rec	F1	Pre	Rec	F1
Ground Truth		31.3	69.1	64.6	21.3	62.9	45.6
OpenSet [12]		44.3	67.2	64.4	14.6	50.1	40.4

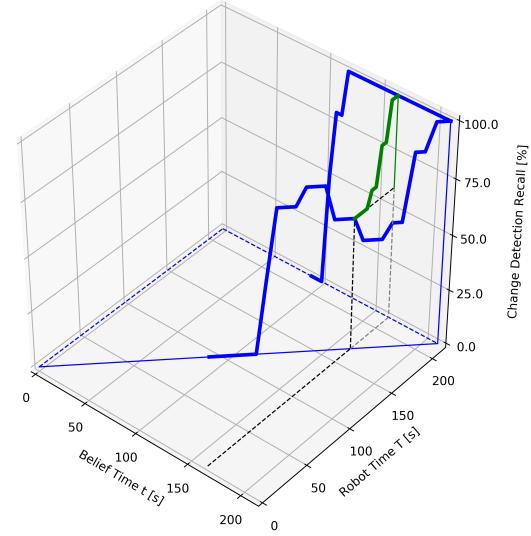


Fig. 7: Spatio-temporal belief about the state of the scene at different times t , given observations of the robot up to its current time T .

primitive image regions given by Segment Anything [20] using CLIP features [36], approximately based on the logic described in [12]. With the different segmentation frontends, we maintain high performance compared to using ground-truth segmentation as input. The main limitation is in terms of whether the segmentation frontend is able to accurately detect the objects that change. The open set method suffers in recall as the "object-ness" of its detections is less well defined, leading to false negatives on object detections. Nonetheless, Khronos is able to filter and extract meaningful objects from the noisy input observations. These findings demonstrate that our approach can build spatio-temporal maps of various "objects" of interest.

D. Spatio-temporal Map Beliefs

To highlight the capability of Khronos to build spatio-temporal map beliefs, we show the change detection recall of Khronos in the office scene in Fig. 7. One axis shows the time T that has elapsed since the robot started exploring the scene. On the other axis, we show the belief about the world at time $t < T$, given the information obtained till T . Importantly, the diagonal axis of $T = t$ represents the real-time axis, i.e., the perceived 'present' of the robot. The first change happens around 90 s, but naturally the robot has not yet observed that object again. The robot then starts observing changes and increasing its recall, while new objects appear and disappear, decreasing the recall. Eventually, the robot closes the loop and detects several changes at the start location. Importantly, our approach propagates this information back through time and updates the robot's belief about earlier times. This is

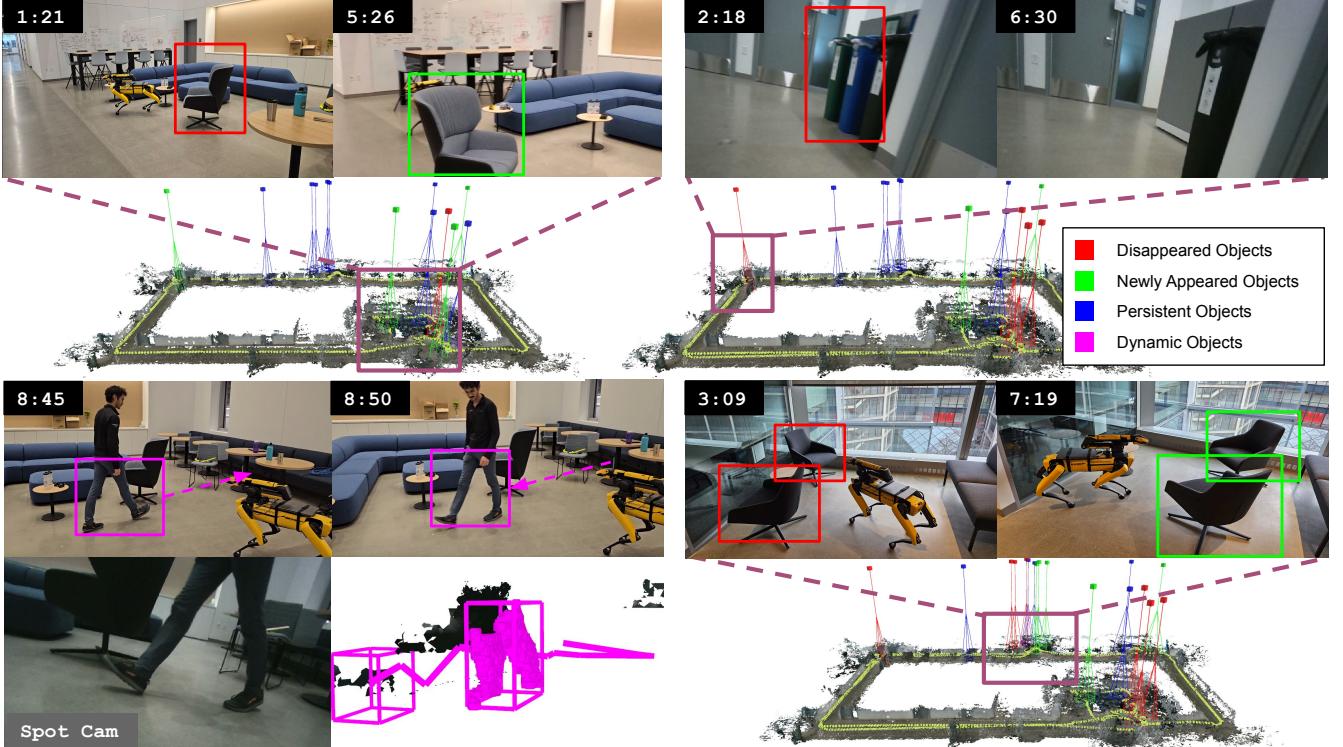


Fig. 8: The spatio-temporal map built with a Boston Dynamics Spot quadruped of an entire floor of a university building. We show a few instances of long term changes in the map: moving a chair to a different location (top left), removing two recycling bins (top right), and changing the arrangement of two chairs (bottom right). The Khroneos reconstruction shown corresponds to the later timestamp, with the detected objects highlighted in the image. Note for example, how the chairs near the window (bottom right, 3:09) are present in the map at 6:30 (top right), but get marked as absent and newly appear in their new pose after the robot revisits at 7:19 (bottom right). We also demonstrate short term dynamics (bottom left) with human action; however, due to the downward facing angle of the camera, Khroneos can only see the legs of the walking human, which are successfully detected.

reflected in the belief time marginal at $T = 220$ s, showing the robot’s estimate of the scene at different times once all data is obtained. A different way to picture this is by looking at a fixed belief time $t = 165$ s, where we can observe how the quality of the robot belief improves as more information is gathered, highlighted in green. Finally, the 4D-metrics (20) can be interpreted as the area under this temporal surface.

E. Mobile Robot Experiments

We validate our approach in two real-world experiments with heterogeneous robots and sensing setups, navigating two different environments. Fig.1 shows qualitative results of the spatio-temporal map produced by Khroneos on a Jackal ground robot in a mezzanine scene.

In this experiment, the robot visits a common area (top right) and then moves around the hallway and through the kitchen. As it revisits the common area, a chair is removed and a cooler newly appears. The robot then performs a longer trajectory, closes the loop back in the common area and correctly detects that the cooler has disappeared again. In addition to long-term changes, the robot also correctly identifies people moving (left) and objects that are not semantically recognized, such as the cart being pushed through the scene (bottom left). It is clear that Khroneos is able to accurately capture both long term object appearances and disappearances, and also short term dynamic movements. In

TABLE III: Performance of Khroneos object detection and long term changes against manually annotated ground truth.

Objects	Changes					
	Pre	Rec	F1	Pre	Rec	F1
86.9	97.7	90.2	87.0	66.7	93.9	

addition, we manually annotate this smaller scale scene with all changes that were induced by the person, and present the results in Tab. III. The observed performance demonstrates that Khroneos is able to accurately capture the objects in the explored scene and reflect the orchestrated changes. Short-term dynamic changes (human motion and a rolling cart in this case) are not reported as manual labeling was too imprecise for the dynamic objects.

Second, we perform experiments on a Boston Dynamic Spot quadruped across the entire floor of a university building. Long-term changes to the scene are orchestrated by removing and adding objects in between robot observations. Short-term dynamics appear in the form of humans walking or moving items in view of the robots. Qualitative results are shown in Fig. 8, where we observe that Khroneos scales well also to this extensive environment and a more dynamic robot platform, and is able to generate a spatio-temporal map that accurately reflects the circumstances of the scene.

F. Computation Time

Finally, it is important for any interactive robot to perceive the scene in real-time with the limited computation available. We therefore present timing results from the large office scene with imperfect odometry. Due to our factorization (16), the active window can operate with approximately constant time complexity. We measure the time to process a frame in the active window at $45.5 \pm 9.2\text{ms}$, resulting in an average frame rate of 22.2 FPS. The scaling of other components of our pipeline is shown in Fig. 9. The top row shows components of the global optimization. The reconstruction of fragments from the active window is marked by occasional spikes, but usually takes $< 1\text{s}$. The deformation of the global background scales linearly with the observed volume, but also generally stays $< 1\text{s}$. The most complex part is solving the optimization problem. This becomes only relevant once loop closures are detected, but requires up to several seconds in the worst case. However, it is important to point out that all of these operations are performed in separate threads and don't need to be performed for every frame, enabling online operation of the robot. Second, the bottom row shows timing of reconciliation components. Here we show worst-time performance, as the ray-hash is recomputed from scratch every time. It is important to point out that the ray hash can also be incrementally allocated and only needs to be recomputed after a large loop closure. Inference on the library of rays for change detection followed by reconciliation is generally quick, taking $< 100\text{ms}$ for all fragments in the scene. These results demonstrate the strong algorithmic properties of the presented factorization, enabling spatio-temporal map building in real-time on mobile robots.

VII. LIMITATIONS

Since Khronos utilizes the bounding-box centroid as the position of a fragment for association edges, accurate fragment associations can be sensitive to partial observations and occlusions. Furthermore, the lack of 6D registration between fragments decreases the effectiveness of global estimation and reconciliation. Adoption of modern object pose and shape estimation and registration techniques [31, 18, 2] would increase the robustness and accuracy of fragment association.

Second, we currently only associate fragments geometrically, meaning that fragments that have moved are not associated. Incorporating fragment descriptors would allow reasoning about the history of moving objects in more detail.

Using a ray-tracing approach for change detection implies that the existence of a reference surface. While we found it to work well in our experiments, change detection in large open spaces with sparse surfaces would perform poorly.

Lastly, we currently keep all object fragments in memory. While this allows for more optimization flexibility, the problem also keeps growing indefinitely, potentially limiting the scalability of Khronos. In an ideal scenario, out-dated or confidently reconciled object fragments are marginalized, such that the number of object fragments stored is proportional to the number of objects to support lifelong mapping missions.

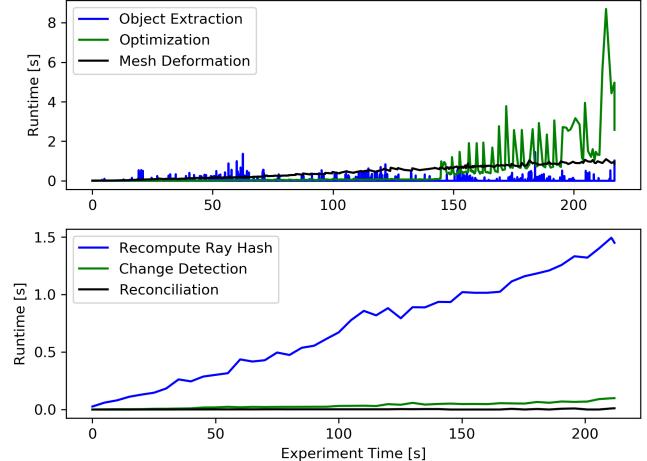


Fig. 9: Run-time of global optimization components (top) and reconciliation components (bottom).

VIII. CONCLUSIONS

In this paper, we defined the SMS problem and presented a novel approach to structure the problem, unifying the tracking of short-term dynamics and the detection of long-term changes in a single formulation. We introduced Khronos, a first metric-semantic spatio-temporal perception system capable of solving the SMS problem and generate a dense 4D spatio-temporal map. We demonstrated that Khronos outperforms recent baselines across metrics pertaining to short and long-term dynamics, can interface with different semantic object formulations, and solve the complex SMS problem in real-time with limited compute. We validated Khronos on several different mobile robotic platforms, showing strong performance in complex real-world environments.

ACKNOWLEDGMENT

We thank Jong Jin Park, Adam Fineberg, and Nathan Hughes for fruitful discussion and Nathan Hughes, Matthew Trang, and Eric Cristofalo for support with the Spot experiments.

REFERENCES

- [1] B. Bescós, J. Fácil, J. Civera, and J. Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters (RA-L)*, 3(4):4076–4083, 2018.
- [2] B. Bescós, C. Campos, J.D. Tardós, and J. Neira. DynaSLAM II: Tightly-coupled multi-object tracking and slam. *IEEE Robotics and Automation Letters (RA-L)*, 6:5191–5198, 2020.
- [3] N. Bore, J. Ekekrantz, P. Jensfelt, and J. Folkesson. Detection and tracking of general movable objects in large three-dimensional maps. *IEEE Transactions on Robotics*, 35(1):231–247, Feb 2019.
- [4] S.L. Bowman, N. Atanasov, K. Daniilidis, and G.J. Pappas. Probabilistic data association for semantic slam. pages 1722–1729, 2017.
- [5] N. Brasch, A. Bozic, J. Lallemand, and F. Tombari. Semantic monocular SLAM for highly dynamic environments. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 393–400, 2018.

- [6] W. Chen, L. Zhu, Y. Guan, C.R. Kube, and H. Zhang. Submap-based pose-graph visual slam: A robust visual exploration and localization system. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 6851–6856, 2018.
- [7] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss. Suma++: Efficient lidar-based semantic slam. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, November 2019.
- [8] L. Cui and C. Ma. SOF-SLAM: A semantic visual SLAM for dynamic environments. *IEEE Access*, 7:166528–166539, 2019.
- [9] F. Dellaert and GTSAM Contributors. borglab/gtsam, May 2022. URL <https://github.com/borglab/gtsam>.
- [10] J. Fu, Y. Du, K. Singh, J.B. Tenenbaum, and J.J. Leonard. Robust change detection based on neural descriptor fields. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, page 2817–2824, October 2022.
- [11] J. Fu, C. Lin, Y. Taguchi, A. Cohen, Y. Zhang, S. Mylabathula, and J.J. Leonard. Planesdf-based change detection for long-term dense mapping. *IEEE Robotics and Automation Letters (RA-L)*, 7(4):9667–9674, Oct 2022.
- [12] Q. Gu, A. Kuwajerwala, S. Morin, K.M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, C. Gan, C.M. de Melo, J.B. Tenenbaum, A. Torralba, F. Shkurti, and L. Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. 2024.
- [13] R. Hachiuma, C. Pirchheim, D. Schmalstieg, and H. Saito. Detectfusion: Detecting and segmenting both known and unknown dynamic objects in real-time slam. In *British Machine Vision Conf. (BMVC)*, 2019.
- [14] M. Henein, J. Zhang, R. Mahony, and V. Ila. Dynamic slam: The need for speed. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, page 2123–2129, May 2020. ICRA.
- [15] D.F. Henning, T. Laidlow, and S. Leutenegger. Bodyslam: Joint camera localisation, mapping, and human motion tracking. In *European Conf. on Computer Vision (ECCV)*, page 656–673, 2022.
- [16] N. Hughes, Y. Chang, S. Hu, R. Talak, R. Abdulhai, J. Strader, and L. Carlone. Foundations of spatial perception for robotics: Hierarchical representations and real-time systems. *Intl. J. of Robotics Research*, 2024.
- [17] J. Jain, J. Li, M. Chiu, A. Hassani, N. Orlov, and H. Shi. OneFormer: One Transformer to Rule Universal Image Segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [18] K.M. Judd and J.D. Gammell. Multimotion visual odometry (mvo). *arXiv preprint arXiv:2110.15169*, 2021.
- [19] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.
- [20] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A.C. Berg, W. Lo, P. Dollár, and R.B. Girshick. Segment anything. In *Intl. Conf. on Computer Vision (ICCV)*, pages 3992–4003, 2023.
- [21] G.S. Krishna, K. Supriya, and S. Baidya. 3DS-SLAM: A 3d object detection based semantic slam towards dynamic indoor environments. 2023.
- [22] E. Langer, T. Patten, and M. Vincze. Robust and efficient object change detection by combining global semantic information and local geometric verification. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 8453–8460, 2020.
- [23] Y. Li, X. Qi, Y. Chen, L. Wang, Z. Li, J. Sun, and J. Jia. Voxel field fusion for 3d object detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [24] R. Long, C. Rauch, T. Zhang, V. Ivan, and S. Vijayakumar. Rigidfusion: Robot localisation and mapping in environments with large dynamic rigid objects. *IEEE Robotics and Automation Letters (RA-L)*, 6(2):3703–3710, April 2021.
- [25] S. Looper, J. Rodriguez-Puigvert, R. Siegwart, C. Cadena, and L. Schmid. 3d vsg: Long-term semantic scene change prediction through 3d variable scene graphs. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 8179–8186. IEEE, 2023.
- [26] D. Maggio, M. Abate, J. Shi, C. Mario, and L. Carlone. Loc-NeRF: Monte carlo localization using neural radiance fields. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2022.
- [27] H. Matsuki, R. Murai, P.H.J. Kelly, and A.J. Davison. Gaussian splatting slam. *arXiv preprint arXiv:2312.06741*, 2023.
- [28] J. McCormac, R. Clark, M. Bloesch, A.J. Davison, and S. Leutenegger. Fusion++: Volumetric object-level SLAM. In *Intl. Conf. on 3D Vision (3DV)*, pages 32–41, 2018.
- [29] B. Mersch, T. Guadagnino, X. Chen, I. Vizzo, J. Behley, and C. Stachniss. Building volumetric beliefs for dynamic environments exploiting map-based moving object segmentation. *IEEE Robotics and Automation Letters (RA-L)*, PP:1–8, 08 2023.
- [30] E. Michael, T. Summers, T.A. Wood, C. Manzie, and I. Shames. Probabilistic data association for semantic slam at scale. pages 4359–4364, 2022.
- [31] J. Morris, Y. Wang, and V. Ila. The importance of coordinate frames in dynamic slam. *arXiv preprint arXiv:2312.04031*, 2023.
- [32] J.-M. Park, J.-H. Jang, S.-M. Yoo, S.-K. Lee, U.-H. Kim, and J.-H. Kim. Changesim: Towards end-to-end online scene change detection in industrial indoor environments. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 8578–8585, Sep 2021.
- [33] J. Qian, V. Chatrath, J. Yang, J. Servos, A.P. Schoellig, and S.L. Waslander. Pocd: Probabilistic object-level change detection and volumetric mapping in semi-static scenes. In *Robotics: Science and Systems (RSS)*, Jul 2022.
- [34] J. Qian, V. Chatrath, J. Servos, A. Mavrinac, W. Burgard, S.L. Waslander, and A.P. Schoellig. Pov-slam: Probabilistic object-aware variational slam in semi-static environments. In *Robotics: Science and Systems (RSS)*, 2023.
- [35] Y. Qiu, C. Wang, W. Wang, M. Henein, and S. Scherer. Airdos: Dynamic slam benefits from articulated objects. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 8047–8053, May 2022.
- [36] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *Intl. Conf. on Machine Learning (ICML)*, 2021.
- [37] Y. Ren, B. Xu, C.L. Choi, and S. Leutenegger. Visual-inertial multi-instance dynamic slam with object-level relocalisation. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 11055–11062, 2022.
- [38] A. Rosinol, M. Abate, Y. Chang, and L. Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2020.
- [39] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone. Kimera: from SLAM to spatial perception with 3D dynamic scene graphs. *Intl. J. of Robotics Research*, 40(12–14):1510–1546, 2021.
- [40] A. Rosinol, J.J. Leonard, and L. Carlone. NeRF-SLAM: Real-time dense monocular SLAM with neural radiance fields. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2023.
- [41] M. Rünz and L. Agapito. Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4471–4478. IEEE, 2017.
- [42] L. Schmid, J. Delmerico, J. Schönberger, J. Nieto, M. Polle-

- fey, R. Siegwart, and C. Cadena. Panoptic multi-TSDFs: a flexible representation for online multi-resolution volumetric mapping and long-term dynamic scene consistency. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 8018–8024, 2022.
- [43] L. Schmid, O. Andersson, A. Sulser, P. Pfreundschuh, and R. Siegwart. Dynablox: Real-time detection of diverse dynamic objects in complex environments. *IEEE Robotics and Automation Letters (RA-L)*, 8(10):6259–6266, 2023.
- [44] W. Shi, J. Xu, D. Zhu, G. Zhang, X. Wang, J. Li, and X. Zhang. Rgb-d semantic segmentation and label-oriented voxelgrid fusion for accurate 3d semantic mapping. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(1):183–197, 2022.
- [45] W. Shi, J. Xu, D. Zhu, G. Zhang, X. Wang, J. Li, and X. Zhang. Meshnet-sp: A semantic urban 3d mesh segmentation network with sparse prior. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(1):183–197, 2022.
- [46] J.C.V. Soares, V. S. Medeiros, G. Fischer Abti, M. Becker, G. Caurin, M. Gattass, and M.A. Meggiolaro. Visual localization and mapping in dynamic and changing environments. *J. of Intelligent and Robotic Systems*, 109(4):95, 2023.
- [47] S. Song, H. Lim, A. J. Lee, and H. Myung. Dynavins: A visual-inertial slam for dynamic environments. *IEEE Robotics and Automation Letters*, 7(4):11523–11530, Oct 2022.
- [48] Z. Song, G. Zhang, J. Xie, L. Liu, C. Jia, S. Xu, and Z. Wang. Voxelnextfusion: A simple, unified and effective voxel fusion framework for multi-modal 3d object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–12, 2023.
- [49] M. Strecke and J. Stueckler. EM-Fusion: Dynamic object-level slam with probabilistic data association. In *Intl. Conf. on Computer Vision (ICCV)*. IEEE, October 2019.
- [50] T. Sun, Y. Hao, S. Huang, S. Savarese, K. Schindler, M. Pollefeys, and I. Armeni. Nothing stands still: A spatiotemporal benchmark on 3d point cloud registration under large geometric and temporal change. 2023.
- [51] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A.J. Davison, and S. Leutenegger. MID-Fusion: Octree-based object-level multi-instance dynamic SLAM. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 5231–5237, 2019.
- [52] B. Xu, A.J. Davison, and S. Leutenegger. Learning to complete object shapes for object-level mapping in dynamic scenes. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, page 2257–2264, October 2022. IROS.
- [53] C. Yan, D. Qu, D. Wang, D. Xu, Z. Wang, B. Zhao, and X. Li. GS-SLAM: Dense visual slam with 3d gaussian splatting, 2024.
- [54] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone. Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. *IEEE Robotics and Automation Letters (RA-L)*, 5(2):1127–1134, 2020.
- [55] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei. DS-SLAM: A semantic visual slam towards dynamic environments. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1168–1174, Oct 2018.
- [56] P. Yu, C. Guo, Y. Liu, and H. Zhang. Fusing semantic segmentation and object detection for visual slam in dynamic scenes. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*, 2021.
- [57] Z. Zhan, W. Jian, Y. Li, and Y. Yue. A slam map restoration algorithm based on submaps and an undirected connected graph. *IEEE Access*, 9:12657–12674, 2021.
- [58] L. Zhu, S. Huang, K. Schindler, and I. Armeni. Living scenes: Multi-object relocalization and reconstruction in changing 3d environments. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.