Dense Incremental Metric-Semantic Mapping for Multi-Agent Systems via Sparse Gaussian Process Regression

Ehsan Zobeidi[1] Alec Koppel[2] Nikolay Atanasov[1]

Abstract—We develop an online probabilistic metric-semantic mapping approach for mobile robot teams relying on streaming RGB-D observations. The generated maps contain full continuous distributional information about the geometric surfaces and semantic labels (e.g., chair, table, wall). Our approach is based on online Gaussian Process (GP) training and inference, and avoids the complexity of GP classification by regressing a truncated signed distance function (TSDF) of the regions occupied by different semantic classes. Online regression is enabled through a sparse pseudo-point approximation of the GP posterior. To scale to large environments, we further consider spatial domain partitioning via an octree data structure with overlapping leaves. An extension to the multi-robot setting is developed by having each robot execute its own online measurement update and then combine its posterior parameters via local weighted geometric averaging with those of its neighbors. This yields a distributed information processing architecture in which the GP map estimates of all robots converge to a common map of the environment while relying only on local one-hop communication. Our experiments demonstrate the effectiveness of the probabilistic metric-semantic mapping technique in 2-D and 3-D environments in both single and multi-robot settings.

I. INTRODUCTION

Autonomous systems navigating and executing complex tasks in real-world environments require an understanding of the 3-D geometry and semantic context of the environment. This paper develops a probabilistic metric-semantic mapping algorithm, using streaming distance and semantic category observations onboard a robot, to reconstruct geometric surfaces and their semantic identity (e.g., chairs, tables, doors) via sparse online GP regression. In addition to a multi-modal environment abstraction, probabilistic metric-semantic mapping provides uncertainty estimates that can aid safe navigation and active mapping algorithms. To support collaboration among multiple robots operating in the same environment, we also consider a distributed setting in which each robot observes the environment locally, with its onboard sensors, and communicates its local map with one-hop neighbor robots to arrive at a common map of the environment observed across the whole robot network.

We focus on a TSDF representation [1], [2] which defines geometric surfaces implicitly, as the zero level-set of a TSDF function. TSDF surface representations have gained popularity due to their high accuracy (compared to regular, adaptive, or
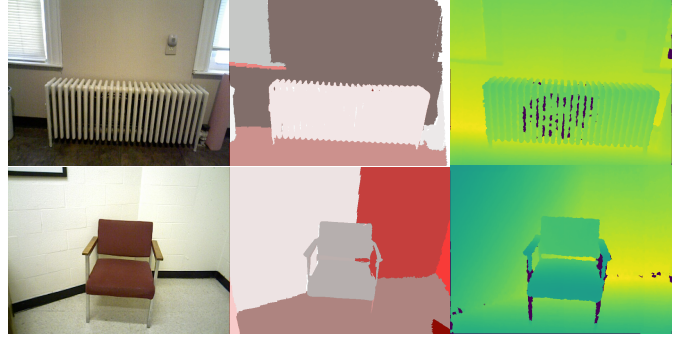

Fig. 1: RGB images (first column), segmented images (second column), and depth images (third column) used by the proposed approach for online construction of dense metric-semantic maps.

sparse grid representations [3], [4]) and ability to directly provide distance and gradient information (compared to explicit mesh representations [5]) useful to specification of safety and visibility constraints. Classification of the geometric surfaces into semantic categories is crucial for context understanding and specification of complex robot tasks [6]–[8]. Many classification techniques, however, provide maximum likelihood, instead of Bayesian, estimates because efficient probabilistic classification remains an open problem in machine learning [9], [10]. The challenge is that discrete data likelihoods are not conjugate with a continuous map prior. While one may employ Laplace approximations to partially mitigate this challenge [11]), we propose a multi-class TSDF inference approach based on Bayesian *regression*.

We employ GP regression [12] to incorporate spatial correlation into a probabilistic resolution-free TSDF map of the 3-D environment. GP inference has been successfully used to obtain continuous map representations [13]–[15] but existing formulations are binary (instead of multi-class) and model occupancy (instead of a distance field). Range sensors, such as Lidars and depth cameras, do not provide direct TSDF observations because they measure distance in a specific viewing direction rather than to the nearest obstacle surface. To obtain TSDF training examples, we triangulate each depth image into a local mesh surface and measure the distance to it from a set of 3-D locations.

Onboard sensors provide repeated observations of the same scene. While this redundancy is important for mitigating measurement noise, the amount of training data keeps growing over time. Hence, an important consideration for metric-semantic mapping is to build maps whose memory and computation requirements are designated by the underlying structure of the environment, rather than the number of distance and category observations. Unfortunately, GP training scales cubi-

[1]Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093, USA {ezobeidi,natanasov}@ucsd.edu
[2]Computational and Information Sciences Directorate, U.S. Army Research Laboratory, Adelphi, MD 20783, USA alec.e.koppel.civ@mail.mil

cally with the number training examples but there are various ways to address this bottleneck [16]–[19]. We observe that, in our setting, the data can be compressed significantly through averaging before GP training and, notably, this does not affect the posterior TSDF distribution. The remaining training pairs are used as *pseudo points* [16] to support the continuous GP representation with a finite set of parameters. To reduce the complexity in large maps further, one might consider local kriging, decomposing the spatial domain into subdomains and making predictions at a test location using only the pseudo points contained within the subdomain. Choosing independent subdomains, however, leads to discontinuities of the predicted TSDF function at the subdomain boundaries. Ensemble methods that construct multiple local estimators and use a weighted combination of their predictions include Bayesian committee machines [20], [21], sparse probabilistic regression [22], or infinite mixtures Gaussian process experts [23]. These techniques avoid the discontinuities of local kriging but their computation cost is still significant for online training. Inspired by the adaptive occupancy representation of Octomap [3], we propose an efficient approach that decomposes the environment into an Octree of overlapping subdomains, while preventing discontinuities in the GP posterior. Combining these ideas yields a hierarchical pseudo-point parameterization of the GP, which may be updated online to achieve incremental probabilistic mapping. Our method generates dense metric-semantic surfaces and, yet, remains efficient even in large environments.

Finally, we provide a distributed formulation of our TSDF GP regression, enabling multiple robots to collaboratively build a common metric-semantic map of the environment. Each robot updates a local GP Octree pseudo-point approximation but synchronizes its pseudo-point statistics by averaging with its one-hop communication neighbors. Our distributed inference approach is inspired by probabilistic consensus techniques [24], [25], but we generalize those from using a fixed parameter dimension to a changing number of pseudo-point parameters, resulting from robots observing new environment regions online. We prove that the local GP estimates of each individual robot converge in *finite time* to the same GP posterior that would have been obtained by a central server using all observations obtained from all robots.

A preliminary version of this work was presented in [26]. This version improves the theoretical development for the centralized single-robot setting and extends the approach to a decentralized multi-robot setting by introducing a novel approach for distributed incremental sparse GP regression with theoretical guarantees for consistent estimation. Additionally, this paper demonstrates the effectiveness of our decentralized approach via evaluations in 2-D simulation and 3-D real data sets. The main **contributions** of this work are to:

- develop an online GP training and inference algorithm for TSDF regression that enables 3-D semantic segmentation of the environment from streaming sensor data,
- ensure controllable computation and memory complexity while providing a continuous-space probabilistic representations of the environment,
- provide a distributed formulation of the TSDF GP regres-

sion, which enables a robot team to collaboratively build a common metric-semantic map from local observations and one-hop communication with provably equivalent quality to batch centralized estimation.

Our metric-semantic mapping approach is demonstrated in simulated and real-world datasets and may be used either offline, with all sensory data provided in advance, or online, processing distance and semantic category observations incrementally as they arrive.

II. RELATED WORK

Various representations have been proposed for occupancy or geometric surface estimation from range or depth measurements. Occupancy grid mapping [27] discretizes the environment into a regular voxel grid and estimates the occupancy probability of each voxel independently. A dense voxel representation quickly becomes infeasible for large domains and adaptive resolution data structures, such as an octree, are necessary [3], [28]. While accurate maps may also be constructed using point cloud [29], [30] or surfel [31], [32] representations, such sparse maps do not easily support collision and visibility checking for motion and manipulation planning. Recent work is considering explicit polygonal mesh [5], [33], [34] and implicit signed distance function [35]–[38] models. We focus our review on TSDF techniques as they are most closely related to our work.

The seminal work of Curless et al. [1] emphasized the representation power of TSDF and showed that dense surface modeling can be done incrementally using range images. KinectFusion [35] achieved online TSDF mapping and RGB-D camera pose estimation by storing weighted TSDF values in a voxel grid and performing multi-scale iterative closest point (ICP) alignment between the predicted surface and the depth images. Niessner et al. [4] demonstrated that TSDF mapping can be achieved without regular or hierarchical grid data structures by hashing TSDF values only at voxels near the surfaces. These three works inspired a lot of subsequent research, allowing mapping of large environments [39], real-time operation without GPU acceleration [40], [41], map correction upon loop closure [42], [43], and semantic category inference [44]. Bylow et al. [45] propose a direct minimization of TSDF projective depth error instead of relaying on explicit data association or downsampling as in ICP. TSDF maps are accurate and collision checking in them is essentially a look-up operation, prompting their use as an alternative to occupancy grids for robot motion planning and collision checking [38], [46]. Voxblox [37] incrementally builds a (non-truncated) Euclidean signed distance field (ESDF), applying a wavefront algorithm to the hashed TSDF values. Fiesta [38] improves the ESDF construction by introducing two independent queues for inserting and deleting obstacles. Saulnier et al. [47] show that weights of the TSDF values arise as the variance of a Kalman filter and may be used as an uncertainty measure for autonomous exploration and active TSDF mapping.

Most TSDF mapping techniques, however, forgo probabilistic representations in the interest of scalability. Gaussian process (GP) inference has been used to capture correlation

in binary occupancy mapping. O'Callaghan et al. [13] is among the first works to apply GP regression to infer a latent occupancy function using data from a range sensor. The GP posterior is squashed to a binary observation model a posteriori to recover occupancy likelihood. The resulting probabilistic least-squares method is more efficient than GP classification but still scales cubically with the amount of training data. To address this, several works [14], [21], [48], [49] rely on sparse kernels to perform separate GP regressions with small subsets of the training data and Bayesian Committee Machines (BCM) to fuse the separate estimates into a full probabilistic occupancy map. Ramos et al. [50] proposed fast kernel approximations to project the occupancy data into a Hilbert space where a logistic regression classifier can distinguish occupied and free space. This idea has been extended to dynamic maps [51] as well as into a variational autoencoder formulation [52] that compresses the local spatial information into a latent low-dimensional feature representation and then decodes it to infer the occupancy of a scene. Guo and Atanasov [53] showed that using a regular grid discretization of the latent function and a decomposable radial kernel leads to special structure of the kernel matrix (kronecker product of Toeplitz matrices) that allows linear time and memory representation of the occupancy distribution.

Augmenting occupancy representations with object and surface category information is an important extension, allowing improved situational awareness and complex mission specification for robots. Several works [7], [54]–[57] employ conditional random fields (CRFs) to capture semantic information. Vineet et al. [54] provide incremental reconstruction and semantic segmentation of outdoor environments using a hash-based voxel map and a mean-field inference algorithm for densely-connected CRFs. These techniques are accurate but also computationally expensive because they operate over each map element. Zheng et al. [58] incorporate spatial information across multiple levels of abstraction and form a probability distribution over semantic attributes and geometric representations of places using TopoNet, a deep sum-product neural network. Grinvald et al. [44] reconstruct individual object shapes from multi-view segmented images and assemble the estimates in a voxelized TSDF map. Gan et al. [59] propose a continuous-space multi-class mapping approach, which relies on a Dirichlet class prior, a Categorical observation likelihood, and Bayesian kernel inference to extrapolate the class likelihoods to continuous space. Rosinol et al. [5], provides a modern perception library by combining the state of the art in geometric and semantic understanding.

In many applications, metric-semantic mapping may be performed by a team of collaborating robots. Relying on centralized estimation has numerous limitations related to the communication, computation, and storage requirements of collecting all robot measurements and map estimates at a central server. It is important to develop distributed techniques that allow local inference and storage at each robot, communication over few-hop neighborhoods, and consensus among the robot estimates. Techniques extending network consensus [60] to distributed probabilistic estimation [**?**], [24], [61]–[63] are closely related. These works show that distributed estimation

of a finite-dimensional parameter is consistent when the probability density functions maintained by different nodes are averaged over one-hop neighborhoods in a strongly connected, potentially time-varying graph. Our work extends these techniques to distributed probabilistic estimation functions relying on local averaging of sparse (pseudo-point) GP distributions. Specific to cooperative semantic mapping, Choudhary et al. [64] develop distributed pose-graph optimization algorithms based on successive and Jacobi over-relaxation to split the computation among the robots. Koch et al. [65] develop a parallel multi-threaded implementation for cooperative 2-D SDF mapping. Lajoie et al. [66] propose a distributed SLAM approach with peer-to-peer communication that rejects spurious inter-robot loop closures using pairwise consistent measurement sets.

III. PROBLEM FORMULATION

Consider a team of n robots, communicating over a network represented as an undirected graph $G = (\mathcal{V}, \mathcal{E})$ with vertices $\mathcal{V} := \{1, ..., n\}$ and edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. An edge $(i, j) \in \mathcal{E}$ from robot i to robot j exists if the two robots can communicate. The robots directly connected to robot i are called *neighbors* and will be denoted by $\mathcal{N}_i := \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$.

The robots operate in an unknown workspace, represented as a subset of Euclidean space, $\mathcal{W} \subset \mathbb{R}^3$. The workspace consists of two disjoint subsets $\mathcal{O}$ and $\mathcal{F}$, comprising obstacles and free space, respectively, i.e., $\mathcal{W} = \mathcal{O} \cup \mathcal{F}$. The obstacle region is a closed set that is a pairwise disjoint union, $\mathcal{O} = \cup_{l=1}^{\mathcal{C}} \mathcal{O}_l$, of $\mathcal{C}$ closed sets, each denoting the region occupied by object instances from the same semantic class. For example, $\mathcal{O}_1$ may be the space occupied by all chairs, while $\mathcal{O}_2$ may be the space occupied by all tables.

Each robot is equipped with a sensor, such as a lidar scanner or an RGB-D camera, that provides distance and class observations of the objects in its vicinity. We assume that the position $\mathbf{p}_t^i \in \mathbb{R}^3$ and orientation $\mathbf{R}_t^i \in SO(3)$ of each sensor $i \in \mathcal{V}$ at time step t are known, e.g., from a localization algorithm running onboard the robots. We model a sensor observation as a set of rays (unit vectors), e.g., corresponding to lidar scan rays or RGB-D image pixels.

Definition 1. A *sensor frame* $\mathbf{E}^i = \{\boldsymbol{\eta}_k^i\}_k$ is a set of vectors $\boldsymbol{\eta}_k^i \in \mathbb{R}^3$ such that $\|\boldsymbol{\eta}_k^i\| = 1, \forall i, k$.

At time t, the k-th sensor ray of robot i, starts at position $\mathbf{p}_t^i$ and has direction $\mathbf{R}_t^i \boldsymbol{\eta}_k^i$. Each ray measures the distance to and semantic class of the object that it intersects with first. In practice, the class measurements are obtained from a semantic segmentation algorithm (e.g., [67]), applied to the RGB image or lidar scan (see Fig. 1), while the distance measurements are provided either as a transformation of the depth image or directly from the lidar scan.

Definition 2. A *sensor observation* of robot i at time t is a collection of distance $\lambda_{t,k}^i \in \mathbb{R}_{\geq 0}$ and object class $c_{t,k}^i \in \{1, ..., \mathcal{C}\}$ measurements acquired along the rays $\boldsymbol{\eta}_k^i \in \mathbf{E}^i$.

We define the relationship among the object sets $\mathcal{O}_l$ and the sensor observations $\lambda_{t,k}^i$, $c_{t,k}^i$ next.

Definition 3. The *truncated signed directional distance function* (TSDDF) $h_l(\mathbf{x}, \boldsymbol{\eta})$ of object class $\mathcal{O}_l$, is the signed distance from $\mathbf{x} \in \mathcal{W}$ to the boundary $\partial \mathcal{O}_l$ in direction $\boldsymbol{\eta} \in \mathbb{R}^3$, truncated to a maximum of $\bar{d} \geq 0$, i.e.,

$$h_l(\mathbf{x}, \boldsymbol{\eta}) := \begin{cases} -\min\left(d_{\boldsymbol{\eta}}(\mathbf{x}, \partial\mathcal{O}_l), \bar{d}\right) & \text{if } \mathbf{x} \in \mathcal{O}_l \\ \min\left(d_{\boldsymbol{\eta}}(\mathbf{x}, \partial\mathcal{O}_l), \bar{d}\right) & \text{if } \mathbf{x} \in \mathcal{W} \setminus \mathcal{O}_l, \end{cases}$$

$$d_{\boldsymbol{\eta}}(\mathbf{x}, \partial\mathcal{O}_l) := \min\left\{d \geq 0 \mid \mathbf{x} + d\boldsymbol{\eta} \in \partial\mathcal{O}_l\right\}. \quad (1)$$

According to Def. 3, $h_l(\mathbf{p}_t^i, \mathbf{R}_t^i \boldsymbol{\eta}_k^i)$ is the (truncated) distance from sensor position $\mathbf{p}_t^i$ to object class $\mathcal{O}_l$ along the direction $\mathbf{R}_t^i \boldsymbol{\eta}_k^i$ of the k-th ray at time t. The class observation $c_{t,k}^i$ is determined by the object set $\mathcal{O}_l$ with minimum absolute TSDDF to $\mathbf{p}_t^i$ along $\mathbf{R}_t^i \boldsymbol{\eta}_k^i$:

$$c_{t,k}^i = \operatorname*{arg\,min}_{l \in \{1,\ldots,\mathcal{C}\}} |h_l(\mathbf{p}_t^i, \mathbf{R}_t^i \boldsymbol{\eta}_k^i)|. \quad (2)$$

The distance observation $\lambda_{t,k}^i$ is a noisy measurement of the distance to the nearest object class:

$$\lambda_{t,k}^i = h_{c_{t,k}^i}(\mathbf{p}_t^i, \mathbf{R}_t^i \boldsymbol{\eta}_k^i) + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma^2), \quad (3)$$

where σ^2 is the variance of the distance measurement noise. These definitions are illustrated in Fig. 2.

Given sensor poses $\mathbf{p}_t^i$, $\mathbf{R}_t^i$ and streaming onboard observations $\lambda_{t,k}^i$, $c_{t,k}^i$ for $t = 1, 2, \ldots$, the main objective of this work is to construct a metric-semantic map of the observed environment online by estimating the object class sets $\mathcal{O}_l = \{\mathbf{x} \in \mathcal{W} \mid \min_{\boldsymbol{\eta}} h_l(\mathbf{x}, \boldsymbol{\eta}) \leq 0\}$, implicitly represented by the TSDDFs $h_l(\mathbf{x}, \boldsymbol{\eta})$. Note that each object class is associated with a posterior distribution over sensor frames $\boldsymbol{\eta}$. To reduce the complexity of estimating TSDDFs, which are defined for arbitrary directions $\boldsymbol{\eta}$, we consider the more usual TSDF model, defined as the minimum of a TSDDF over $\boldsymbol{\eta}$.

Definition 4. The *truncated signed distance function* (TSDF) $f_l(\mathbf{x})$ of object class $\mathcal{O}_l$ is the truncated signed distance from $\mathbf{x} \in \mathcal{X}$ to the boundary $\partial\mathcal{O}_l$, i.e.,

$$f_l(\mathbf{x}) := h_l(\mathbf{x}, \boldsymbol{\eta}^*) \text{ where } \boldsymbol{\eta}^* = \operatorname*{arg\,min}_{\boldsymbol{\eta}} |h_l(\mathbf{x}, \boldsymbol{\eta})|. \quad (4)$$

We develop incremental sparse Gaussian Process regression to maintain distributions $\mathcal{GP}(\mu_{t,l}^i(\mathbf{x}), k_{t,l}^i(\mathbf{x}, \mathbf{x}'))$ over the TSDF functions $f_l(\mathbf{x})$ in (4) at each robot i, conditioned on the sensor observations $\left\{\lambda_{\tau,k}^i, c_{\tau,k}^i\right\}$ up to time t. We propose a new data compression technique in Sec. IV and apply it in the design of the GP training algorithm for probabilistic TSDF inference in Sec. V. Our approach generates a continuous-space probabilistic model of the distance to and semantic classes of the environment surfaces. To achieve scalable online mapping of large domains, we train independent sparse GP models over an octree cover of the 3-D space.

Next, we extend our approach from a centralized single-robot to a distributed multi-robot formulation. We develop new techniques for distributed incremental sparse GP regression in Sec. VI and apply them to the collaborative semantic TSDF mapping problem in Sec. VII. Our method allows each robot to update its own sparse TSDF GP model, relying on local sensor observations and one-hop information exchange with its neighborhoods, yet guarantees theoretically that the model
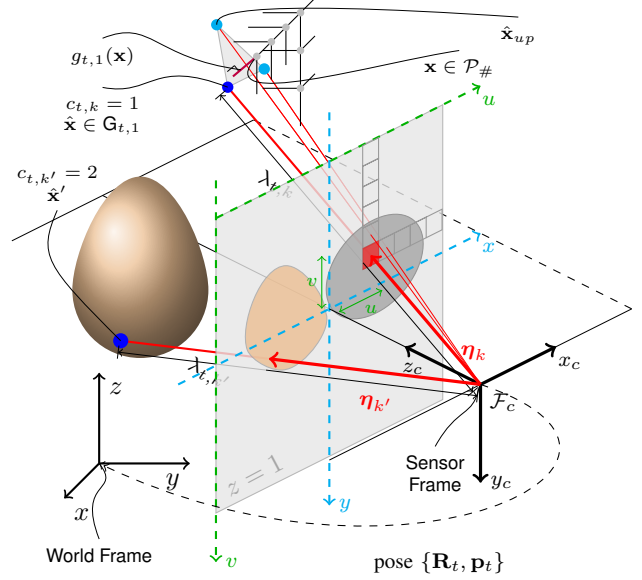

Fig. 2: Sensor observation at time t showing the distance $\lambda_{t,k}$, $\lambda_{t,k'}$ and class $c_{t,k}$, $c_{t,k'}$ measurements obtained along sensors rays $\boldsymbol{\eta}_k$, $\boldsymbol{\eta}_k' \in \mathbf{E}$ when a camera sensor is at position $\mathbf{p}_t$ with orientation $\mathbf{R}_t$. The pseudo points $\mathcal{P}_\#$ (see Sec. V-A) close to the observed surface are shown in gray.

parameters of different robots converge in finite-time to the same parameters that would be obtained by centralized GP regression. The effectiveness of our approach is demonstrated in single- and multi-robot experiments using simulated 2-D data in Sec. VIII and real 3-D data in Sec. IX.

IV. DATA COMPRESSION FOR INCREMENTAL SPARSE GAUSSIAN PROCESS REGRESSION

This section reviews sparse Gaussian Process regression and introduces a new approach for compressing training data acquired by repeated observation of the same locations, which is typical when an onboard robot sensor observes the same environment multiple times. Our data compression allows training a GP model with much fewer samples, yet provably generates the same GP posterior that would have been computed using the full uncompressed training set. Finally, the sparse GP model and the data compression allow us to design an efficient incremental GP algorithm that updates the GP posterior with sequential data instead of recomputing it from scratch.

A. Background on Sparse GP Regression

A Gaussian Process is a set of random variables such that the joint distribution of any finite subset of them is Gaussian. A GP-distributed function $f(\mathbf{x}) \sim \mathcal{GP}(\mu_0(\mathbf{x}), k_0(\mathbf{x}, \mathbf{x}'))$ is defined by a mean function $\mu_0(\mathbf{x})$ and a covariance (kernel) function $k_0(\mathbf{x}, \mathbf{x}')$. The mean and covariance are such that for any finite set $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$, the random vector $f(\mathcal{X}) := [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_M)]^\top \in \mathbb{R}^M$ has mean with j-th element $\mu_0(\mathbf{x}_j)$ and covariance matrix with (j,l)-th element $k_0(\mathbf{x}_j, \mathbf{x}_l)$ for $j, l = 1, \ldots, M$. Given a training set $\mathcal{D} = \{(\mathbf{x}_j, y_j)\}_{j=1}^M$,

generated according to $y_j = f(\mathbf{x}_j) + \eta_j$ with independent Gaussian noise $\eta_j \sim \mathcal{N}(0, \sigma^2)$, the posterior distribution of the random function $f(\mathbf{x})$ can be obtained from the joint distribution of the value $f(\mathbf{x})$ at an arbitrary location $\mathbf{x}$ and the random vector $\mathbf{y} := [y_1, \ldots, y_M]^\top$ of measurements. In detail, the joint distribution is:

$$\begin{bmatrix} f(\mathbf{x}) \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_0(\mathbf{x}) \\ \mu_0(\mathcal{X}) \end{bmatrix}, \begin{bmatrix} k_0(\mathbf{x}, \mathbf{x}) & k_0(\mathbf{x}, \mathcal{X}) \\ k_0(\mathcal{X}, \mathbf{x}) & k_0(\mathcal{X}, \mathcal{X}) + \sigma^2 I \end{bmatrix} \right),$$

while the corresponding conditional distribution $f(\mathbf{x})|\mathcal{X}, \mathbf{y} \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ has mean and covariance functions:

$$\mu(\mathbf{x}) := \mu_0(\mathbf{x}) + k_0(\mathbf{x}, \mathcal{X})(k_0(\mathcal{X}, \mathcal{X}) + \sigma^2 I)^{-1}(\mathbf{y} - \mu_0(\mathcal{X})),$$
$$k(\mathbf{x}, \mathbf{x}') := k_0(\mathbf{x}, \mathbf{x}') - k_0(\mathbf{x}, \mathcal{X})(k_0(\mathcal{X}, \mathcal{X}) + \sigma^2 I)^{-1} k_0(\mathcal{X}, \mathbf{x}').$$
(5)

Computing the GP posterior has cubic complexity in the number of observations M due to the matrix inversion in (5).

Inspired by Snelson and Ghahramani [16], we introduce a sparse approximation to the GP posterior in (5) using a set of *pseudo points* $\mathcal{P} \subset \mathcal{D}$ whose number $|\mathcal{P}| \ll M$. The key idea is to first determine the distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ of $\mathbf{f} := f(\mathcal{P})$ conditioned on $\mathcal{X}, \mathbf{y}$ according to (5):

$$\boldsymbol{\mu} := \mu_0(\mathcal{P}) + k_0(\mathcal{P}, \mathcal{X})(k_0(\mathcal{X}, \mathcal{X}) + \sigma^2 I)^{-1}(\mathbf{y} - \mu_0(\mathcal{X}))$$
$$= \mu_0(\mathcal{P}) + k_0(\mathcal{P}, \mathcal{P})(k_0(\mathcal{P}, \mathcal{P}) + \Gamma)^{-1}\boldsymbol{\gamma} \quad (6)$$

$$\Sigma := k_0(\mathcal{P}, \mathcal{P}) - k_0(\mathcal{P}, \mathcal{X})(k_0(\mathcal{X}, \mathcal{X}) + \sigma^2 I)^{-1} k_0(\mathcal{X}, \mathcal{P}),$$
$$= k_0(\mathcal{P}, \mathcal{P})(k_0(\mathcal{P}, \mathcal{P}) + \Gamma)^{-1} k_0(\mathcal{P}, \mathcal{P})$$

where $\Gamma := k_0(\mathcal{P}, \mathcal{X})(\Lambda + \sigma^2 I)^{-1} k_0(\mathcal{X}, \mathcal{P})$, $\Lambda := k_0(\mathcal{X}, \mathcal{X}) - k_0(\mathcal{X}, \mathcal{P})k_0(\mathcal{P}, \mathcal{P})^{-1}k_0(\mathcal{P}, \mathcal{X})$, and $\boldsymbol{\gamma} := k_0(\mathcal{P}, \mathcal{X})(\Lambda + \sigma^2 I)^{-1}(\mathbf{y} - \mu_0(\mathcal{X}))$. Using the definitions of information matrix $\Omega := \Sigma^{-1}$ and information mean $\boldsymbol{\omega} := \Omega\boldsymbol{\mu}$, we can equivalently write:

$$\boldsymbol{\omega} = \Omega\mu_0(\mathcal{P}) + k_0(\mathcal{P}, \mathcal{P})^{-1}\boldsymbol{\gamma},$$
$$\Omega = k_0(\mathcal{P}, \mathcal{P})^{-1}(k_0(\mathcal{P}, \mathcal{P}) + \Gamma)k_0(\mathcal{P}, \mathcal{P})^{-1}. \quad (7)$$

Then, the posterior density of $f(\mathbf{x})$ conditioned on $\mathcal{X}, \mathbf{y}$ is:

$$p(f(\mathbf{x})|\mathcal{X}, \mathbf{y}) = \int p(f(\mathbf{x})|\mathbf{f})p(\mathbf{f}|\mathcal{X}, \mathbf{y})d\mathbf{f} \quad (8)$$

which is a GP with mean and covariance functions:

$$\mu(\mathbf{x}) = \mu_0(\mathbf{x}) + k_0(\mathbf{x}, \mathcal{P})k_0(\mathcal{P}, \mathcal{P})^{-1}\left(\Omega^{-1}\boldsymbol{\omega} - \mu_0(\mathcal{P})\right)$$
$$k(\mathbf{x}, \mathbf{x}') = k_0(\mathbf{x}, \mathcal{P})k_0(\mathcal{P}, \mathcal{P})^{-1}\Omega^{-1}k_0(\mathcal{P}, \mathcal{P})^{-1}k_0(\mathcal{P}, \mathbf{x}')$$
$$+ k_0(\mathbf{x}, \mathbf{x}') - k_0(\mathbf{x}, \mathcal{P})k_0(\mathcal{P}, \mathcal{P})^{-1}k_0(\mathcal{P}, \mathbf{x}'). \quad (9)$$

If we assume that conditioned on $\mathcal{P}$, the measurements y_j are generated independently, i.e., Λ is approximated by a diagonal matrix with elements $\lambda(\mathbf{x}_j) := k_0(\mathbf{x}_j, \mathbf{x}_j) - k_0(\mathbf{x}_j, \mathcal{P})k_0(\mathcal{P}, \mathcal{P})^{-1}k_0(\mathcal{P}, \mathbf{x}_j)$, then the complexity of computing $\boldsymbol{\mu}, \Sigma$ in (6) (training) and $\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')$ in (9) (testing) are $O(|\mathcal{P}|^2|\mathcal{X}| + |\mathcal{P}|^3)$ and $O(|\mathcal{P}|^2)$, respectively, instead of $O(|\mathcal{X}|^3)$ and $O(|\mathcal{X}|^2)$ without pseudo points in (5). The use of pseudo points leads to significant computational savings when $|\mathcal{P}| \ll |\mathcal{X}|$. We assume that the kernel parameters are optimized offline and focus on online computation of the terms in (9), needed for prediction.

B. Repeated Input Data Compression

Next, we detail a way to obtain additional savings in terms of data storage requirements. Specifically, if the training data $\mathcal{D} = (\mathcal{X}, \mathbf{y})$ contains repeated observations from the same locations, i.e., the points in $\mathcal{X}$ are not unique, then the GP training complexity can be reduced from cubic in $|\mathcal{X}|$ to cubic in the number of distinct points in $\mathcal{X}$. We formalize this in the following proposition, which establishes that the GP posterior is unchanged if we compress the observations in $\mathbf{y}$ obtained from the same locations in $\mathcal{X}$.

Proposition 1. *Consider $f(\mathbf{x}) \sim \mathcal{GP}(\mu_0(\mathbf{x}), k_0(\mathbf{x}, \mathbf{x}'))$. Let:*

$$\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_2, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_n\}$$
$$\mathbf{y} = [y_{1,1}, \ldots, y_{1,m_1}, y_{2,1}, \ldots, y_{2,m_2}, \ldots, y_{n,1}, \ldots, y_{n,m_n}]^\top$$

be data generated from the model $y_{i,j} = f(\mathbf{x}_i) + \eta_{i,j}$ with $\eta_{i,j} \sim \mathcal{N}(0, \sigma^2)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m_i$. Let:

$$\mathcal{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}, \ \boldsymbol{\zeta} = \left[\frac{1}{m_1}\sum_{j=1}^{m_1} y_{1,j}, \ldots, \frac{1}{m_n}\sum_{j=1}^{m_n} y_{n,j} \right]^\top$$
(10)

be a compressed version of the data generated from $f(\mathbf{x}_i)$ with noise $\hat{\eta}_i \sim \mathcal{N}(0, \frac{\sigma^2}{m_i})$. Then, $f(\mathbf{x})|\mathcal{X}, \mathbf{y}$ and $f(\mathbf{x})|\mathcal{P}, \boldsymbol{\zeta}$ have the same Gaussian Process distribution $\mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ with:

$$\mu(\mathbf{x}) = \mu_0(\mathbf{x}) + k_0(\mathbf{x}, \mathcal{P})Z(\boldsymbol{\zeta} - \mu_0(\mathcal{P})),$$
$$k(\mathbf{x}, \mathbf{x}') = k_0(\mathbf{x}, \mathbf{x}') - k_0(\mathbf{x}, \mathcal{P})Zk_0(\mathcal{P}, \mathbf{x}'), \quad (11)$$

where $Z^{-1} := k_0(\mathcal{P}, \mathcal{P}) + \sigma^2 \operatorname{diag}(\mathbf{m})^{-1}$ and $\mathbf{m}$ is a vector with elements m_i.

Proof. The distribution of $f(\mathbf{x})|\mathcal{X}, \mathbf{y}$ is provided in (5). Using the data $\mathcal{P}, \boldsymbol{\zeta}$, instead of $\mathcal{X}, \mathbf{y}$, to compute the posterior GP distribution of $f(\mathbf{x})$, according to (5), leads to the expression in (11). We need to show that (5) and (11) are equal given the relationship between $\mathcal{X}, \mathbf{y}$ and $\mathcal{P}, \boldsymbol{\zeta}$ in (10). Let E be a binary matrix defined such that $k_0(\mathcal{X}, \mathbf{x}) = Ek_0(\mathcal{P}, \mathbf{x})$. Note that $k_0(\mathcal{X}, \mathcal{X}) = Ek_0(\mathcal{P}, \mathcal{P})E^\top$, $k_0(\mathbf{x}, \mathcal{X}) = k_0(\mathbf{x}, \mathcal{P})E^\top$, $E^\top E = \operatorname{diag}(\mathbf{m})$, and $\boldsymbol{\zeta} = (E^\top E)^{-1}E^\top \mathbf{y}$. Using these expressions in (5) leads to:

$$\mu(\mathbf{x}) = \mu_0(\mathbf{x}) + $$
$$k_0(\mathbf{x}, \mathcal{P})E^\top(Ek_0(\mathcal{P}, \mathcal{P})E^\top + \sigma^2 I)^{-1}(\mathbf{y} - E\mu_0(\mathcal{P})),$$
$$k(\mathbf{x}, \mathbf{x}') = k_0(\mathbf{x}, \mathbf{x}') - \quad (12)$$
$$k_0(\mathbf{x}, \mathcal{P})E^\top(Ek_0(\mathcal{P}, \mathcal{P})E^\top + \sigma^2 I)^{-1}Ek_0(\mathcal{P}, \mathbf{x}').$$

An application of the matrix inversion lemma followed by algebraic manipulation shows that $E^\top(Ek_0(\mathcal{P}, \mathcal{P})E^\top + \sigma^2 I)^{-1} = (k_0(\mathcal{P}, \mathcal{P}) + \sigma^2(E^\top E)^{-1})^{-1}(E^\top E)^{-1}E^\top = Z(E^\top E)^{-1}E^\top$. Replacing this and $\boldsymbol{\zeta} = (E^\top E)^{-1}E^\top \mathbf{y}$ in (12) shows that the GP distributions of $f(\mathbf{x})|\mathcal{X}, \mathbf{y}$ and $f(\mathbf{x})|\mathcal{P}, \boldsymbol{\zeta}$ are equal. □

Prop. 1 allows us to summarize a training set $\mathcal{X}, \mathbf{y}$ by keeping the distinct points $\mathcal{P} \subset \mathcal{X}$ as well as the average observation value $\zeta(\mathbf{p})$ and number of times $m(\mathbf{p})$ that each point $\mathbf{p} \in \mathcal{P}$ has been observed. Given these statistics, the mean function $\mu(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ of the posterior GP can be obtained according to (11) with

$\zeta := \zeta(\mathcal{P})$ and $\mathbf{m} := m(\mathcal{P})$. When the training points $\mathcal{X}$ contain many repetitions, the subset $\mathcal{P}$ of distinct points is a natural choice of pseudo points (Sec. IV-A) and, in this case, the posterior obtained from training with $\mathcal{P}$ is *exact* (Prop. 1) instead of an approximation of the posterior obtained from training with $\mathcal{X}$. We exploit this compression technique for efficient incremental GP training when the same observations are observed multiple times.

C. Incremental Compressed Sparse GP Regression

Suppose now that, instead of a single training set $\mathcal{D}$, the data are provided sequentially, i.e., an additional dataset $\tilde{\mathcal{D}}_t$ of points $\tilde{\mathcal{X}}_t$ with labels $\tilde{\mathbf{y}}_t$ is provided at each time step t. The cumulative data up to time t are $\mathcal{D}_t := \cup_{\tau=1}^t \tilde{\mathcal{D}}_\tau$. Based on Prop. 1, we can define an incrementally growing set of pseudo points $\mathcal{P}_t$ with associated number of observations $m_t(\mathbf{p})$ and average observation $\zeta_t(\mathbf{p})$ for $\mathbf{p} \in \mathcal{P}_t$ and observation precision Z_t. We show how to update these statistics when a new dataset $\tilde{\mathcal{D}}_{t+1} = (\tilde{\mathcal{X}}_{t+1}, \tilde{\mathbf{y}}_{t+1})$ arrives at time $t+1$. Let $\tilde{\mathcal{P}}_{t+1}$ be the set of unique points in $\tilde{\mathcal{X}}_{t+1}$ with number of observations $\tilde{m}_{t+1}(\mathbf{p})$ and average observation $\tilde{\zeta}_{t+1}(\mathbf{p})$ for $\mathbf{p} \in \tilde{\mathcal{P}}_{t+1}$. The update of $\mathcal{P}_t$, $m_t(\mathbf{p})$ and $\zeta_t(\mathbf{p})$ is:

$$\mathcal{P}_{t+1} = \mathcal{P}_t \cup \tilde{\mathcal{P}}_{t+1}$$

$$m_{t+1}(\mathbf{p}) = \begin{cases} m_t(\mathbf{p}) + \tilde{m}_{t+1}(\mathbf{p}), & \text{if } \mathbf{p} \in \mathcal{P}_t, \\ \tilde{m}_{t+1}(\mathbf{p}), & \text{else}, \end{cases} \quad (13)$$

$$\zeta_{t+1}(\mathbf{p}) = \begin{cases} \frac{m_t(\mathbf{p})\zeta_t(\mathbf{p}) + \tilde{m}_{t+1}(\mathbf{p})\tilde{\zeta}_{t+1}(\mathbf{p})}{m_{t+1}(\mathbf{p})}, & \text{if } \mathbf{p} \in \mathcal{P}_t, \\ \tilde{\zeta}_{t+1}(\mathbf{p}), & \text{else}. \end{cases}$$

To update the observation precision Z_t, first consider the existing pseudo points $\mathcal{P}_t$. Let l be the index of $\mathbf{p} \in \mathcal{P}_t$ in Z_t. Define $\epsilon_l := \sigma^2 \left(\frac{1}{m_{t+1}(\mathbf{p})} - \frac{1}{m_t(\mathbf{p})} \right)$, $B_0 := Z_t$, and for $l = 1, \ldots, |\mathcal{P}_t|$:

$$B_{l+1} = \left(B_l^{-1} + \epsilon_l \mathbf{e}_l \mathbf{e}_l^\top \right)^{-1} = B_l - \frac{B_l \mathbf{e}_l \mathbf{e}_l^\top B_l}{\frac{1}{\epsilon_l} + \mathbf{e}_l^\top B_l \mathbf{e}_l}. \quad (14)$$

With some abuse of notation, let $B := B_{|\mathcal{P}_t|}$ be the observation precision after all $\mathbf{p} \in \mathcal{P}_t$ have been updated. Finally, we update B by introducing the pseudo points $\tilde{\mathcal{P}}_{t+1} \setminus \mathcal{P}_t$ that have been observed for the first time:

$$Z_{t+1} = \begin{bmatrix} B^{-1} & C \\ C^\top & D \end{bmatrix}^{-1} = \begin{bmatrix} B + BCSC^\top B & -BCS \\ -SC^\top B & S \end{bmatrix}, \quad (15)$$

where $C := k_0(\mathcal{P}_t, \tilde{\mathcal{P}}_{t+1} \setminus \mathcal{P}_t)$, $D := k_0(\tilde{\mathcal{P}}_{t+1} \setminus \mathcal{P}_t, \tilde{\mathcal{P}}_{t+1} \setminus \mathcal{P}_t) + \sigma^2 \operatorname{diag}(\tilde{m}_{t+1}(\tilde{\mathcal{P}}_{t+1} \setminus \mathcal{P}_t))^{-1}$, and $S := (D - C^\top BC)^{-1}$. By recursively tracking these matrix inverses, the posterior update can be executed efficiently every time a new observation arrives with complexity that is cubic in the number of new distinct points. This is a significant improvement over naïve GP training.

Unfortunately, this complexity still exhibits computational bottlenecks over large domains, where the number of pseudo points $\mathcal{P}_t$ continues to grow with t. Returning to the TSDF mapping problem, this situation happens when a robot continuously explores a large 3-D environment. We introduce an octree spatial decomposition with overlapping subregions,

allowing us to train independent GPs with a fixed maximum number of pseudo points in each subregion. This aspect, as well as how the training sets are constructed from the robot observations, discussed in Sec. III, and utilized for probabilistic semantic TSDF mapping are the focus of the following section.

V. Probabilistic Metric-Semantic Mapping

In this section, we consider the single-robot mapping problem. For simplicity of notation, we suppress the superscript i that denotes the robot index. The sensor measurements $\{\lambda_{t,k}, c_{t,k}\}$ are generated according to the models in (2) and (3) that depend on the TSDDFs $\{h_l(\mathbf{x}, \boldsymbol{\eta})\}$ of the different semantic classes in the environment. As mentioned in Sec. III, instead of $\{h_l(\mathbf{x}, \boldsymbol{\eta})\}$, we focus on estimating the TSDFs $\{f_l(\mathbf{x})\}$, whose domains are lower-dimensional. We apply the incremental GP regression technique developed in Sec. IV. Since the sensor data $\{\lambda_{t,k}, c_{t,k}\}$ are not direct samples from the TSDFs, they need to be transformed into training sets $\tilde{\mathcal{D}}_{t,l}$, suitable for updating the GP distributions of $\{f_l(\mathbf{x})\}$.

A. Training Set Construction

The class measurements allow us to associate the sensor data with particular semantic classes, while the distance measurements allow us to estimate the points where the sensor rays hit the object sets $\mathcal{O}_l$. We define the following point sets for each detected semantic class at time t:

$$\mathsf{G}_{t,l} = \{\hat{\mathbf{x}} \in \mathbb{R}^3 \,|\, \hat{\mathbf{x}} = \lambda_{t,k} \mathbf{R}_t \boldsymbol{\eta}_k + \mathbf{p}_t \text{ and } c_{t,k} = l\}. \quad (16)$$

The values $f_l(\hat{\mathbf{x}})$ of the TSDFs are close to zero at points $\hat{\mathbf{x}} \in \mathsf{G}_{t,l}$ because the sensor rays hit an object surface close to these locations.

As shown in Prop. 1, the complexity of online GP training can be improved by forcing the training data to repeatedly come from a finite set of points. We choose a grid discretization $\mathcal{P}_\#$ of the workspace $\mathcal{W}$ and construct a training set by selecting points $\mathbf{x} \in \mathcal{P}_\#$, that are at most $\epsilon > 0$ away from the points $\hat{\mathbf{x}} \in \mathsf{G}_{t,l}$, and approximating their TSDF values $f_l(\mathbf{x}) \approx g_{t,l}(\mathbf{x})$ (see Fig. 2). Precisely, the training data sets are constructed at time t as:

$$\tilde{\mathcal{D}}_{t,l} = \{(\mathbf{x}, g_{t,l}(\mathbf{x})) | \mathbf{x} \in \mathcal{P}_\#, \exists \hat{\mathbf{x}} \in \mathsf{G}_{t,l} \text{ s.t. } ||\mathbf{x} - \hat{\mathbf{x}}||_2 \leq \epsilon\}. \quad (17)$$

In the case of a camera sensor, the TSDF value $g_{t,l}(\mathbf{x})$ of a pseudo point $\mathbf{x}$ is obtained by projecting $\mathbf{x}$ to the image plane and approximating its distance from the distance values of nearby pixels. In detail, suppose $\boldsymbol{\eta}_k$ is the unit vector corresponding to the pixel closest to the projection of $\mathbf{x}$ (red pixel in Fig. 2) and let $\hat{\mathbf{x}} \in \mathsf{G}_{t,l}$ be the coordinates of its ray endpoint (blue point in Fig. 2). Let $\hat{\mathbf{x}}_{right}$ and $\hat{\mathbf{x}}_{up}$ (two cyan points in Fig. 2) be the ray endpoints of two adjacent pixels. Then, $g_{t,l}(\mathbf{x})$ is the signed distance from $\mathbf{x}$ to the plane defined by $\hat{\mathbf{x}}$, $\hat{\mathbf{x}}_{right}$, and $\hat{\mathbf{x}}_{up}$:

$$g_{t,l}(\mathbf{x}) := \mathbf{n}^\top(\mathbf{x} - \hat{\mathbf{x}}), \quad \mathbf{n} := \operatorname{sign}(\mathbf{q}^\top(\mathbf{p}_t - \hat{\mathbf{x}}))\mathbf{q},$$

$$\mathbf{q} = \frac{(\hat{\mathbf{x}}_{right} - \hat{\mathbf{x}}) \times (\hat{\mathbf{x}}_{up} - \hat{\mathbf{x}})}{||(\hat{\mathbf{x}}_{right} - \hat{\mathbf{x}}) \times (\hat{\mathbf{x}}_{up} - \hat{\mathbf{x}})||}, \quad (18)$$

where $\mathbf{q}$ is the normal of the plane and the signed distance from $\mathbf{p}_t$ to the plane is positive because the sensor is known to be outside of the object set $\mathcal{O}_l$. With the input variables as distance observations and target variables as truncated signed distance field specified, we shift to how actually compute the posterior inference.

B. Incremental TSDF Inference

Recall that we are using streaming measurements to update the GP distributions of the TSDFs $\{f_l(\mathbf{x})\}$. We derived an incremental sparse GP update in Sec. IV-C. Here, we use the transformed TSDF training data $\tilde{\mathcal{D}}_{t,l}$ to update the GP distribution for each class l. At time t, the new data are $\tilde{\mathcal{D}}_{t,l} = (\tilde{\mathcal{X}}_{t,l}, \tilde{\mathbf{y}}_{t,l})$ and the new pseudo points are $\tilde{\mathcal{P}}_{t,l} = \tilde{\mathcal{X}}_{t,l} \setminus \mathcal{P}_{t-1,l}$. Given $\tilde{\mathcal{X}}_{t,l}, \tilde{\mathbf{y}}_{t,l}, \tilde{\mathcal{P}}_{t,l}$ for each class l, we can update $\mathcal{P}_{t,l}, \zeta_{t,l},$ $m_{t,l}$ via (13). If online prediction is required, we can also update the precision matrix $Z_{t,l}$ using (14) and (15). Then, we have the GPs of all classes updated, and can predict the TSDF at any query point according to (11). Next we discuss how the inferred posterior may be employed to construct a semantic category prediction.

C. Semantic Category Prediction

Next, we discuss how to predict the semantic class labels on the surfaces of the implicitly estimated object sets $\mathcal{O}_l$. While we did not explicitly model noise in the class observations in (2), in practice, semantic segmentation algorithms may produce incorrect pixel-level classification. This leads to some sensor observations $\lambda_{t,k}, c_{t,k}$ being incorrectly included into the training set $\hat{\mathcal{D}}_{t,l}$ of a different semantic class. This happens, for example, if objects from two different classes, say l_1 and l_2, are spatially close to each other and, in an RGB image, parts of the boundary of one are classified as belonging to the other class. Over time, with multiple sensor observations, the TSDF approximations for both classes l_1 and l_2 may contain pseudo points $\mathbf{x} \in \mathcal{P}_\#$ with small TSDF values, indicating an object surface at the same location. To predict the correct semantic class, we compare the likelihoods of the different classes at surface points using the posterior GP distributions of the TSDFs $f_l(\mathbf{x})$.

Proposition 2. *Let $\mathcal{GP}(\mu_{t,l}(\mathbf{x}), k_{t,l}(\mathbf{x}, \mathbf{x}'))$ be the distributions of the truncated signed distance functions $f_l(\mathbf{x})$ at time t, determined according to (11). Consider an arbitrary point $\mathbf{x} \in \partial\mathcal{O}$ on the surface of the obstacle set, i.e., $\mathbf{x}$ is such that $f_l(\mathbf{x}) = 0$ for some class $l \in \{1, \ldots, \mathcal{C}\}$. Then, the probability that the true class label of $\mathbf{x}$ is $c \in \{1, \ldots, \mathcal{C}\}$ is:*

$$\mathbb{P}\left(\arg\min_l |f_l(\mathbf{x})| = c \;\middle|\; \min_l |f_l(\mathbf{x})| = 0\right) = \frac{\frac{1}{\sigma_{t,c}(\mathbf{x})}\phi\left(\frac{\mu_{t,c}(\mathbf{x})}{\sigma_{t,c}(\mathbf{x})}\right)}{\sum_l \frac{1}{\sigma_{t,l}(\mathbf{x})}\phi\left(\frac{\mu_{t,l}(\mathbf{x})}{\sigma_{t,l}(\mathbf{x})}\right)},$$

where $\phi(\cdot)$ is the probability density function of the standard normal distribution and $\sigma_{t,l}(\mathbf{x}) := \sqrt{k_{t,l}(\mathbf{x}, \mathbf{x})}$.
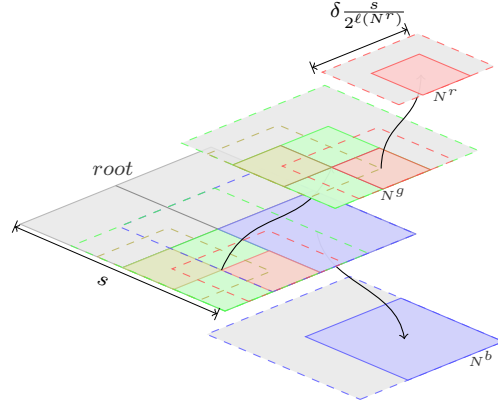

Fig. 3: Illustration of an octree data structure, containing two pseudo points (blue and cyan) in two dimensions. The support regions $\mathcal{S}(\cdot)$ and test regions $\mathcal{T}(\cdot)$ of three nodes N^r, N^g, N^b are shown as dashed and filled areas with red, green, and blue color, respectively. No pseudo points are contained in the test region $\mathcal{T}(N^g)$ (filled green) of node N^g but two pseudo points are in its support region $\mathcal{S}(N^g)$ (dashed green). In this example, the maximum number of allowable pseudo points for each region is $max(N) = 1$, so node N^g is split into the red (N^r) and yellow (not labeled) regions. The cyan pseudo point belongs to both $\mathcal{P}_{t,l}(N^b)$ and $\mathcal{P}_{t,l}(N^r)$.

Proof.

Let $l_c(z) := \mathbb{P}\left(\arg\min_l |f_l(\mathbf{x})| = c \text{ and } \min_l |f_l(\mathbf{x})| \le |z|\right)$.
Since $\mathbb{P}\left(\min_l |f_l(\mathbf{x})| \le |z|\right) = \sum_l l_l(z)$:

$$\mathbb{P}\left(\arg\min_l |f_l(\mathbf{x})| = c \;\middle|\; \min_l |f_l(\mathbf{x})| \le |z|\right) = \frac{l_c(z)}{\sum_l l_l(z)}$$

The term we are interested in computing is $\lim_{z \to 0} \frac{l_c(z)}{\sum_l l_l(z)}$. Let $\mathbf{x}$ be an arbitrary (test) point and define $\mu_l := \mu_{t,l}(\mathbf{x})$ and $\sigma_l := \sigma_{t,l}(\mathbf{x})$ for $l = 1, \ldots, \mathcal{C}$. The GP distribution of f_l stipulates that its value at $\mathbf{x}$ has a density function $p(z) = \frac{1}{\sigma_l}\phi\left(\frac{z - \mu_l}{\sigma_l}\right)$. Hence, $\mathbb{P}(|f_l(\mathbf{x})| \ge z) = 1 - \Phi\left(\frac{|z| - \mu_l}{\sigma_l}\right) + \Phi\left(\frac{-|z| - \mu_l}{\sigma_l}\right)$. Note that $l_c(z)$ corresponds to the probability that $|f_c(\mathbf{x})| \le |f_l(\mathbf{x})|$ for all l. Since all f_l are independent of each other:

$$l_c(z) = \frac{1}{\sigma_c}\int_{-z}^{z}\phi\left(\frac{\zeta - \mu_c}{\sigma_c}\right)\prod_{l \neq c}\left(1 - \Phi\left(\frac{|\zeta| - \mu_l}{\sigma_l}\right) + \Phi\left(\frac{-|\zeta| - \mu_l}{\sigma_l}\right)\right)d\zeta$$

The claim is concluded by $\lim_{z \to 0}\frac{l_c(z)}{2z} = \frac{1}{\sigma_c}\phi\left(\frac{-\mu_c}{\sigma_c}\right)$. $\qquad\square$

The class distribution for an arbitrary point $\mathbf{x} \in \mathcal{W}$, not lying on an object surface, may also be obtained, as shown in the proof of Prop. 2 but is both less efficient to compute and rarely needed in practice.

D. Octree of Gaussian Processes

Even after compressing the TSDF training data using Prop. 1 to a small set of distinct pseudo points, the GP training complexity still scales cubically with the number of pseudo points. To ensure that online training is possible for large environments, we develop an octree data structure with overlapping octant regions to store the pseudo points. We train independent GPs in each of these regions, which is efficient

since the maximum number of pseudo points per region is fixed. The region overlap serves to eliminate discontinuities in the resulting TSDF estimate. At test time, the TSDF value of a query point is inferred using only the parameters of the corresponding region according to (11). The overlapping octant regions are illustrated in Fig. 3.

Formally, an octree of pseudo points is a tree data structure such that each internal node has at most eight children. Each node is associated with a spatial region in the 3-D workspace $\mathcal{W}$. The root is associated with a cube with side length $s > 0$, which is recursively subdivided into up to eight overlapping octant regions by the eight child nodes. Each node N maintains the following information:

1) $\ell(N) \geq 0$: level of N in the tree, starting from 0 at the root node.
2) $ctr(N) \in \mathbb{R}^3$: center of the region associated with N.
3) $\mathcal{S}(N) := \{\mathbf{x} \in \mathcal{W} | \ \|\mathbf{x} - ctr(N)\|_\infty \leq \delta \frac{s}{2^{\ell(N)+1}}\}$: support region of N with $\delta > 1$.
4) $\mathcal{T}(N) := \{\mathbf{x} \in \mathcal{W} | \ \|\mathbf{x} - ctr(N)\|_\infty \leq \frac{s}{2^{\ell(N)+1}}\}$: test region of N.
5) $\mathcal{P}(N) \subseteq \mathcal{S}(N) \cap \mathcal{P}_\#$: set of pseudo points assigned to this node
6) $max(N)$: node N splits into eight children if the number of observed pseudo points $\mathcal{P}(N)$ exceeds $max(N)$
7) $children(N)$: empty set if N is a leaf and, otherwise, a set of eight nodes at level $\ell(N) + 1$ with centers in $\{ctr(N) + s_x\mathbf{e}_1 + s_y\mathbf{e}_2 + s_z\mathbf{e}_3 | s_x, s_y, s_z \in \{-\frac{s}{2^{\ell(N)+1}}, +\frac{s}{2^{\ell(N)+1}}\}\}$.

The pseudo points $\mathcal{P}_{t,l}$ observed up to time t (see Sec. V-B) are stored in octree data structures for each class l. The points assigned to node N for class l at time t are $\mathcal{P}_{t,l}(N) := \mathcal{P}_{t,l} \cap \mathcal{S}(N)$. The pseudo points $\mathcal{P}_{t,l}(N)$ of each leaf node N are used to train an independent GP. At time step t, prediction associated with each class l for test points in the region $\mathcal{T}(N)$ is performed by the GP associated with node N. The class distribution of test points with small predicted TSDF values (surface points) is determined according to Prop. 2. With the data structure developed for efficient representations of possibly large domains, we next shift to how the proposed incremental posterior inference scheme may be decentralized across a collection of interconnected robots.

VI. DISTRIBUTED INCREMENTAL SPARSE GP REGRESSION

In this section, we develop a distributed version of the incremental sparse GP regression in Sec. IV. We consider n robots, communicating over a network $G = (\mathcal{V}, \mathcal{E})$. Each robot $i \in \mathcal{V}$ receives its own local observations $\tilde{\mathcal{D}}_t^i = (\tilde{\mathcal{X}}_t^i, \tilde{\mathbf{y}}_t^i)$ at time t and extracts newly observed pseudo points $\tilde{\mathcal{P}}_t^i$, with associated number of observations $\tilde{\mathbf{m}}_t^i$ and average values $\tilde{\boldsymbol{\zeta}}_t^i$, as detailed in Sec. IV-C. This information is used to update the complete set of pseudo points $\mathcal{P}_t^i$ observed up to time t, along with the number of observations $\mathbf{m}_t^i$ and average values $\boldsymbol{\zeta}_t^i$, according to (13). These parameters $\Theta_t^i := \{\mathcal{P}_t^i, \mathbf{m}_t^i, \boldsymbol{\zeta}_t^i\}$, maintained by robot i, define a complete GP distribution for the function $f(\mathbf{x})$, with mean and covariance functions in (11).

While each robot may estimate $f(\mathbf{x})$ individually, we consider how the robots may exchange information to estimate $f(\mathbf{x})$ collaboratively. Our approach is inspired by techniques extending network consensus [60] to distributed probabilistic estimation [24], [25], [61]–[63]. We observe that the continuous-space GP distribution of $f(\mathbf{x})$ is induced by the statistics $\mathbf{m}_t^i$, $\boldsymbol{\zeta}_t^i$ associated with the finite number of pseudo points $\mathcal{P}_t^i$ and, hence, if the robots exchange information about and agree on these finite-dimensional parameters, then the corresponding GP distributions of $f(\mathbf{x})$ at each robot will agree. Our main innovation is a distributed algorithm for updating the sparse GP parameters of one robot using the parameters of its one-hop neighbors' distributions. While existing results apply to fixed finite-dimensional parameter estimation, our approach applies to function estimation with an infinite-dimensional GP distribution, updated via consensus on an incrementally growing set of pseudo-point parameters.

In Sec. IV-A, we demonstrated a duality between the joint Gaussian distribution over the pseudo points and the posterior GP induced by these pseudo points. Specifically, if the joint Gaussian distribution of the pseudo points in (6) or (7) is available, then we can calculate the mean and covariance functions the GP in (9). This observation suggests that it is sufficient to keep track of the information mean and information matrix of the joint Gaussian distribution of the pseudo points.

Before continuing, we define a few key quantities related to the graph G. Specifically, denote as $A \in \mathbb{R}^{n \times n}$ its adjacency matrix, whose elements A_{ij} may be non-binary. Let $D := \mathrm{diag}(D_{11}, \ldots, D_{nn})$ be the diagonal degree matrix of the graph with elements $D_{ii} = \sum_{j \neq i} A_{ij}$ and $L := D - A$ be the graph Laplacian. Define a weight matrix $W := I - \nu L$ for $0 \leq \nu \leq \frac{1}{\Delta}$, where $\Delta = \max(D_{11}, \ldots, D_{nn})$ is the maximum node degree. The vector of ones, $\mathbf{1} \in \mathbb{R}^n$, is an eigenvector of W since $L\mathbf{1} = \mathbf{0}$. Also, W is a row-stochastic nonnegative and primitive matrix [60] and, hence, has a stationary distribution, specified by its left eigenvector $\boldsymbol{\pi}$ with $\sum_{i=1}^n \pi_i = 1$. This Perron weight matrix construction is common in consensus and distributed gradient descent algorithms [60], [68], [69].

To gain intuition about the construction of consensus schemes over GP posteriors, we first review distributed Kalman filtering for fixed-dimensional parameter estimation.

Remark 1 (Directed time-varying graphs). For simplicity, we consider an undirected static graph G with a fixed weight matrix W. Relying on consensus results for switching networks [60], [70], [71], our results may be generalized to directed and time-varying graphs assuming that the graph sequence is uniformly strongly connected, i.e., there exists an integer $T > 0$ such that the union of the edges over any time interval of length T is strongly connected.

A. Distributed Kalman Filtering

Suppose that the robots aim to estimate a fixed (finite-dimensional) vector $\mathbf{f}$ cooperatively using local observations $\mathbf{y}_t^i$, generated according to a linear Gaussian model:

$$\mathbf{y}_t^i = H^i \mathbf{f} + \boldsymbol{\eta}_t^i, \qquad \boldsymbol{\eta}_t^i \sim \mathcal{N}(0, V^i). \tag{19}$$

Assume that the observations $\mathbf{y}_t^i$ received by robot i are independent over time and from the observations of all other

robots. Assume also that the graph G is connected and that $\mathbf{f}$ is observable if one has access to the observations received by all robots, i.e., the matrix $\begin{bmatrix} H^1 & \cdots & H^n \end{bmatrix}$ has rank equal to the dimension of $\mathbf{f}$. Since individual observations $\mathbf{y}_t^i$ alone may be insufficient to estimate $\mathbf{f}$, the robots need to exchange information. We suppose that each robot starts with a prior probability density function $p_0^i(\mathbf{f})$ over the unknown vector $\mathbf{f}$ and updates it over time, relying on its local observations $\mathbf{y}_t^i$ as well as communication with one-hop neighbors in G.

Rahnama Rad and Tahbaz-Saleh [61] developed a consistent distributed estimation algorithm, in which each agent i uses standard Bayesian updates with its local observations $\mathbf{y}_{t+1}^i$ but, instead of its own prior p_t^i, each agent uses a weighted geometric average of its neighbors' priors:

$$ p_{t+1}^i(\mathbf{f}) \propto p^i(\mathbf{y}_{t+1}^i|\mathbf{f}) \prod_{i=1}^n (p_t^i(\mathbf{f}))^{W_{ij}}, \qquad (20) $$

where $p^i(\mathbf{y}_{t+1}^i|\mathbf{f})$ is an observation model, such as (19), that should satisfy certain regularity conditions [61]. Atanasov et al. [62] showed that if the prior distributions p_0^i are Gaussian and the observation models are linear Gaussian as in (19), the resulting distributed Kalman filter is mean-square consistent (the estimates $\arg\max_{\mathbf{f}} p_t^i(\mathbf{f})$ of all agents i converge in mean square to the true $\mathbf{f}$). Specifically, if the priors are $\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}_0^i, \Sigma_0^i)$ with information matrix $\Omega_0^i := (\Sigma_0^i)^{-1}$ and information mean $\boldsymbol{\omega}_0^i := \Omega_0^i \boldsymbol{\mu}_0^i$, the Gaussian version of the distributed estimator in (20) is:

$$ \boldsymbol{\omega}_{t+1}^i = \sum_{i=1}^n W_{ij} \boldsymbol{\omega}_t^j + H^{i\top} V^{i-1} \mathbf{y}_{t+1}^i $$
$$ \Omega_{t+1}^i = \sum_{j=1}^n W_{ij} \Omega_t^j + H^{i\top} V^{i-1} H^i \qquad (21) $$

because geometric averaging and Bayesian updates with Gaussian densities lead to a Gaussian posterior density [62]. The relationship between geometric means being used for belief propagation in (20) and weighted averaging via mixing matrix W forms the conceptual basis for message passing in the more general GP posterior inference setting which we detail next.

B. Distributed Incremental Sparse GP Regression

The distributed estimation algorithm in (21) does not directly apply to GP regression because the estimation target $f(\mathbf{x})$ is infinite-dimensional. However, the sparse GP regression, described in Sec. IV, relies on a finite (albeit incrementally growing) set of pseudo points $\mathcal{P}_t$, and we show that it is possible to obtain distributed incremental sparse GP regression based on (21). As discussed in the beginning of this section, each robot i maintains parameters $\Theta_t^i := \{\mathcal{P}_t^i, \mathbf{m}_t^i, \boldsymbol{\zeta}_t^i\}$ based on its local observations $\tilde{\mathcal{D}}_t^i = (\tilde{\mathcal{X}}_t^i, \tilde{\mathbf{y}}_t^i)$. Our key idea is to perform weighted geometric averaging over local posteriors, which translates to simple weighted averaging of the means and covariances in (6) of f at a finite set of pseudo points $\mathcal{Q} \supseteq \mathcal{P}_t^i$, which will be specified precisely below. The parameters Θ_t^i maintained by robot i induce a GP distribution over f in (11), which in turn provides a Gaussian probability density function $p_t^i(\mathbf{f}) := p(\mathbf{f}|\Theta_t^i)$ over the (finite-dimensional)

vector $\mathbf{f} := f(\mathcal{Q})$ with mean and covariance, obtained from (11):

$$ \mu_t^i(\mathcal{Q}) := \mu_0^i(\mathcal{Q}) + k_0^i(\mathcal{Q}, \mathcal{P}_t^i) Z_t^i \left(\boldsymbol{\zeta}_t^i - \mu_0^i(\mathcal{P}_t^i) \right), $$
$$ \Sigma_t^i(\mathcal{Q}) := k_0^i(\mathcal{Q}, \mathcal{Q}) - k_0^i(\mathcal{Q}, \mathcal{P}_t^i) Z_t^i k_0^i(\mathcal{P}_t^i, \mathcal{Q}), \qquad (22) $$

where $Z_t^i = (k_0^i(\mathcal{P}_t^i, \mathcal{P}_t^i) + \sigma^2 \operatorname{diag}(\mathbf{m}_t^i)^{-1})^{-1}$. In order to derive decentralized updates for GPs akin to (21), we first present the iterative updates associated with robots' local posteriors in terms of their information mean and information matrix corresponding to the mean and covariance of $p_t^i(\mathbf{f})$.

Lemma 1. *The information mean $\omega_t^i(\mathcal{Q}) := \Omega_t^i(\mathcal{Q})\mu_t^i(\mathcal{Q})$ and information matrix $\Omega_t^i(\mathcal{Q}) := (\Sigma_t^i(\mathcal{Q}))^{-1}$ of the Gaussian probability density function $p_t^i(\mathbf{f}) := p(\mathbf{f}|\Theta_t^i)$ of $\mathbf{f} := f(\mathcal{Q})$ with parameters $\Theta_t^i := \{\mathcal{P}_t^i, \mathbf{m}_t^i, \boldsymbol{\zeta}_t^i\}$ and mean and covariance in (22) are:*

$$ \omega_t^i(\mathcal{Q}) = k_0^i(\mathcal{Q}, \mathcal{Q})^{-1} \mu_0^i(\mathcal{Q}) + \sigma^{-2} \operatorname{diag}(m_t^i(\mathcal{Q})) \zeta_t^i(\mathcal{Q}) $$
$$ \Omega_t^i(\mathcal{Q}) = k_0^i(\mathcal{Q}, \mathcal{Q})^{-1} + \sigma^{-2} \operatorname{diag}(m_t^i(\mathcal{Q})), \qquad (23) $$

where, similar to Sec. IV-C, $m_t^i(\mathbf{p})$ and $\zeta_t^i(\mathbf{p})$ denote the number of observations and average observation, respectively, for $\mathbf{p} \in \mathcal{P}_t^i$ and their domains have been extended to $\mathcal{Q} \supseteq \mathcal{P}_t^i$ by defining $m_t^i(\mathbf{q}) = \zeta_t^i(\mathbf{q}) = 0$ for $\mathbf{q} \in \mathcal{Q} \setminus \mathcal{P}_t^i$.

Proof. Similar to the proof of Prop. 1, let E be a binary matrix such that $k_0^i(\mathcal{P}_t^i, \mathbf{x}) = E k_0^i(\mathcal{Q}, \mathbf{x})$, i.e., E selects the points from the superset $\mathcal{Q}$ which correspond to $\mathcal{P}_t^i$. Note that $k_0^i(\mathcal{Q}, \mathcal{P}_t^i) = k_0^i(\mathcal{Q}, \mathcal{Q}) E^\top$, $k_0^i(\mathcal{P}_t^i, \mathcal{Q}) = E k_0^i(\mathcal{Q}, \mathcal{Q})$, and $k_0^i(\mathcal{P}_t^i, \mathcal{P}_t^i) = E k_0^i(\mathcal{Q}, \mathcal{Q}) E^\top$. The expression for $\Omega_t^i(\mathcal{Q})$ follows from the matrix inversion lemma applied to the covariance matrix in (22) and noting that $E^\top \operatorname{diag}(\mathbf{m}_t^i) E = \operatorname{diag}(m_t^i(\mathcal{Q}))$. Then, note that:

$$ \Omega_t^i(\mathcal{Q}) k_0^i(\mathcal{Q}, \mathcal{P}_t^i) Z_t^i $$
$$ = \left(I + \sigma^{-2} E^\top \operatorname{diag}(\mathbf{m}_t^i) E k_0^i(\mathcal{Q}, \mathcal{Q}) \right) E^\top Z_t^i \qquad (24) $$
$$ = \sigma^{-2} E^\top \operatorname{diag}(\mathbf{m}_t^i) (Z_t^i)^{-1} Z_t^i = \sigma^{-2} E^\top \operatorname{diag}(\mathbf{m}_t^i). $$

Thus, the information mean is:

$$ \omega_t^i(\mathcal{Q}) = \Omega_t^i(\mathcal{Q}) \left(\mu_0^i(\mathcal{Q}) + k_0^i(\mathcal{Q}, \mathcal{P}_t^i) Z_t^i \left(\boldsymbol{\zeta}_t^i - \mu_0^i(\mathcal{P}_t^i) \right) \right) $$
$$ = \Omega_t^i(\mathcal{Q}) \mu_0^i(\mathcal{Q}) + \sigma^{-2} E^\top \operatorname{diag}(\mathbf{m}_t^i) \left(\boldsymbol{\zeta}_t^i - \mu_0^i(\mathcal{P}_t^i) \right) $$
$$ = k_0^i(\mathcal{Q}, \mathcal{Q})^{-1} \mu_0^i(\mathcal{Q}) + \sigma^{-2} E^\top \operatorname{diag}(\mathbf{m}_t^i) \boldsymbol{\zeta}_t^i \qquad (25) $$
$$ = k_0^i(\mathcal{Q}, \mathcal{Q})^{-1} \mu_0^i(\mathcal{Q}) + \sigma^{-2} \operatorname{diag}(m_t^i(\mathcal{Q})) \zeta_t^i(\mathcal{Q}) \qquad \square $$

With the expression for the parametric updates associated with the posterior inference defined by observations acquired locally at robot i only, we next detail how to augment this update with neighboring robots' information.

1) Distributed updates with a fixed pseudo-point set: To begin, suppose that the pseudo-point sets are fixed across all robots, i.e., $\mathcal{P} \equiv \mathcal{P}_t^i$, and the local observations $\tilde{\mathcal{D}}_{t+1}^i = (\tilde{\mathcal{X}}_{t+1}^i, \tilde{\mathbf{y}}_{t+1}^i)$ satisfy $\tilde{\mathcal{X}}_{t+1}^i \subseteq \mathcal{P}$ for all t, i. Then, the information means and matrices in (23) have equal dimensions across the robots, and we can apply the update in (21) directly:

$$ \boldsymbol{\omega}_{t+1}^i = \sum_{i=1}^n W_{ij} \boldsymbol{\omega}_t^j + H_{t+1}^{i\top} (\sigma^2 I)^{-1} \tilde{\mathbf{y}}_{t+1}^i, $$
$$ \Omega_{t+1}^i = \sum_{j=1}^n W_{ij} \Omega_t^j + H_{t+1}^{i\top} (\sigma^2 I)^{-1} H_{t+1}^i, \qquad (26) $$

where $H_{t+1}^i := k_0^i(\tilde{\mathcal{X}}_{t+1}^i, \mathcal{P}) k_0^i(\mathcal{P}, \mathcal{P})^{-1}$, $\boldsymbol{\omega}_t^i := \omega_t^i(\mathcal{P})$, and $\Omega_t^i := \Omega_t^i(\mathcal{P})$. The information means and matrices have a simple structure, and, similar to (13), it is sufficient to track only the number of observations $\mathbf{m}_t^i$ and the average observations $\boldsymbol{\zeta}_t^i$ over time:

$$\boldsymbol{\omega}_{t+1}^i = \sum_{i=1}^n W_{ij} \boldsymbol{\omega}_0^j + \frac{1}{\sigma^2} \sum_{i=1}^n W_{ij} \operatorname{diag}(\mathbf{m}_t^j) \boldsymbol{\zeta}_t^j + \frac{1}{\sigma^2} \operatorname{diag}(\tilde{\mathbf{m}}_{t+1}^i) \tilde{\boldsymbol{\zeta}}_{t+1}^i$$

$$\Omega_{t+1}^i = \sum_{j=1}^n W_{ij} \Omega_0^j + \frac{1}{\sigma^2} \sum_{i=1}^n W_{ij} \operatorname{diag}(\mathbf{m}_t^j) + \frac{1}{\sigma^2} \operatorname{diag}(\tilde{\mathbf{m}}_{t+1}^i), \quad (27)$$

where $\tilde{\mathbf{m}}_{t+1}^i$ and $\tilde{\boldsymbol{\zeta}}_{t+1}^i$ are the number of new observations and new observation averages received by robot i of the pseudo points $\mathcal{P}$ at time $t+1$. We consider the case with incrementally growing pseudo-point sets that are potentially different across the robots before presenting the final distributed update equations for $\mathbf{m}_t^i$ and $\boldsymbol{\zeta}_t^i$. This is the focus of the following subsection.

2) Distributed updates with dynamic pseudo-point sets: Consider the general case where each robot maintains its own pseudo-point set $\mathcal{P}^i$ and the observations $\tilde{\mathcal{D}}_{t+1}^i = (\tilde{\mathcal{X}}_{t+1}^i, \tilde{\mathbf{y}}_{t+1}^i)$ may introduce new pseudo-points $\tilde{\mathcal{P}}_{t+1}^i \not\subseteq \mathcal{P}_t^i$. Our key observation is that the parameters Θ_t^i induce a GP distribution over the whole function f and, hence, can be used to obtain a Gaussian distribution over a pseudo-point set that is larger than $\mathcal{P}_t^i$ according to (22) and (23). Note that the structure of the information mean and information matrix in (23) remains unchanged when the set of pseudo-points $\mathcal{Q}$ changes. To increment the pseudo-point set of robot i at time t, we aggregate the pseudo points $\mathcal{P}_t^j$ of its neighbors and the newly observed pseudo points $\tilde{\mathcal{P}}_{t+1}^i$ as follows:

$$\mathcal{P}_{t+1}^i = \bigcup_{j \in \mathcal{N}_i \cup \{i\}} \mathcal{P}_t^j \cup \tilde{\mathcal{P}}_{t+1}^i \quad (28)$$

Then, the distributed averaging in (21) can be performed over the information means and information matrices in (23) with $\mathcal{Q} = \mathcal{P}_{t+1}^i$ and $H_{t+1}^i := k_0^i(\tilde{\mathcal{X}}_{t+1}^i, \mathcal{P}_{t+1}^i) k_0^i(\mathcal{P}_{t+1}^i, \mathcal{P}_{t+1}^i)^{-1}$:

$$\omega_{t+1}^i(\mathcal{P}_{t+1}^i) = \sum_{i=1}^n W_{ij} \omega_t^j(\mathcal{P}_{t+1}^i) + H_{t+1}^{i\top}(\sigma^2 I)^{-1} \tilde{\mathbf{y}}_{t+1}^i,$$

$$\Omega_{t+1}^i(\mathcal{P}_{t+1}^i) = \sum_{i=1}^n W_{ij} \Omega_t^j(\mathcal{P}_{t+1}^i) + H_{t+1}^{i\top}(\sigma^2 I)^{-1} H_{t+1}^i. \quad (29)$$

We may rewrite the preceding expressions in terms of the number of observations $m_{t+1}^i(\mathbf{p})$ and average observations $\zeta_{t+1}^i(\mathbf{p})$ for any $\mathbf{p} \in \mathcal{P}_{t+1}^i$, akin to (13), by following the steps in (27) for the dynamic pseudo-point case, leading to:

$$m_{t+1}^i(\mathbf{p}) = \sum_{j \in \mathcal{N}_i \cup \{i\}} W_{ij} m_t^j(\mathbf{p}) + \tilde{m}_{t+1}^i(\mathbf{p}), \quad (30)$$

$$\zeta_{t+1}^i(\mathbf{p}) = \frac{\sum_{j \in \mathcal{N}_i \cup \{i\}} W_{ij} m_t^j(\mathbf{p}) \zeta_t^j(\mathbf{p}) + \tilde{m}_{t+1}^i(\mathbf{p}) \tilde{\zeta}_{t+1}^i(\mathbf{p})}{m_{t+1}^i(\mathbf{p})}.$$

With the updates for robot i in terms of its local observations and message passing with its neighbors $\mathcal{N}_i$ specified, we shift in the following subsection to establishing its statistical properties.

C. Theoretical Guarantee for Consistent Estimation

We show that the proposed distributed incremental sparse GP regression defined by (28), (30), and (11) converges to a centralized sparse GP regression, which uses the observation data $\cup_t \cup_i \tilde{\mathcal{D}}_t^i$ from all robots. At each time step t, the centralized estimator receives data $\cup_i \tilde{\mathcal{D}}_t^i$, and, as discussed in Sec. IV-C, updates a global set of pseudo points $\mathcal{P}_t^{ctr}$, the number of times $m_t^{ctr}(\mathbf{p})$ each pseudo point $\mathbf{p} \in \mathcal{P}_t^{ctr}$ has been observed, and the average observation $\zeta_t^{ctr}(\mathbf{p})$ of $\mathbf{p} \in \mathcal{P}_t^{ctr}$. In order to show that the GP maintained by each robot i eventually agrees with the centralized GP, the centralized estimator should also be affected by the Perron weight matrix W. If $W = \frac{1}{n} \mathbf{1} \mathbf{1}^\top$, the information provided by different robots is equally credible and the centralized estimator can use the combined set of observations $\cup_i \tilde{\mathcal{D}}_t^i$ directly. If, however, the left eigenvector $\boldsymbol{\pi}$ of W is not $\mathbf{1}$, then its elements π_i specify different credibility for the different robots. More precisely, the centralized estimator should treat the measurements $\tilde{\mathcal{D}}_t^i$ of robot i as if they were generated with noise variance σ^2 / π_i, instead of the true noise variance σ^2. This is equivalent to scaling the number of observations $\tilde{m}_t^i$ provided by robot i by its "credibility" π_i, leading to the following update for the centralized sparse GP regression parameters:

$$\mathcal{P}_{t+1}^{ctr} = \cup_{i=1}^n \tilde{\mathcal{P}}_{t+1}^i \cup \mathcal{P}_t^{ctr},$$

$$m_{t+1}^{ctr}(\mathbf{p}) = m_t^{ctr}(\mathbf{p}) + \sum_{i=1}^n \pi_i \tilde{m}_{t+1}^i(\mathbf{p}), \quad (31)$$

$$\zeta_{t+1}^{ctr}(\mathbf{p}) = \frac{m_t^{ctr}(\mathbf{p}) \zeta_t^{ctr}(\mathbf{p}) + \sum_{i=1}^n \pi_i \tilde{m}_{t+1}^i(\mathbf{p}) \tilde{\zeta}_{t+1}^i(\mathbf{p})}{m_{t+1}^{ctr}(\mathbf{p})},$$

for all $\mathbf{p} \in \mathcal{P}_{t+1}^{ctr}$. The next result shows that the individual GP distributions maintained by each robot using the distributed updates in (30) converge to the centralized GP distribution determined by the parameters above.

Proposition 3. *Let $\tilde{\mathcal{D}}_t^i = (\tilde{\mathcal{X}}_t^i, \tilde{\mathbf{y}}_t^i)$ be the data received by robot i at time t, associated with pseudo points $\tilde{\mathcal{P}}_t^i \subset \mathcal{P}_\#$ and number of observations $\tilde{m}_t^i(\mathbf{p})$ and average observation $\tilde{\zeta}_t^i(\mathbf{p})$ for $\mathbf{p} \in \mathcal{P}_\#$. If the data streaming stops at some time $T < \infty$, then as $t \to \infty$, the distributions $\mathcal{GP}(\mu_t^i(\mathbf{x}), k_t^i(\mathbf{x}, \mathbf{x}'))$ maintained by each robot i, specified according to (11) with parameters $\mathcal{P}_t^i$, $m_t^i(\mathbf{p})$, $\zeta_t^i(\mathbf{p})$ in (28) and (30) converge to the distribution $\mathcal{GP}(\mu_t^{ctr}(\mathbf{x}), k_t^{ctr}(\mathbf{x}, \mathbf{x}'))$ of the centralized estimator with parameters $\mathcal{P}_t^{ctr}$, $m_t^{ctr}(\mathbf{p})$, $\zeta_t^{ctr}(\mathbf{p})$ in (31), i.e., $|\mu_t^i(\mathbf{x}) - \mu_t^{ctr}(\mathbf{x})| \to 0$ and $|k_t^i(\mathbf{x}, \mathbf{x}') - k_t^{ctr}(\mathbf{x}, \mathbf{x}')| \to 0$ almost surely for all $i \in \mathcal{V}$, $\mathbf{x}, \mathbf{x}'$.*

Proof. Since the distributions $\mathcal{GP}(\mu_t^i(\mathbf{x}), k_t^i(\mathbf{x}, \mathbf{x}'))$ and $\mathcal{GP}(\mu_t^{ctr}(\mathbf{x}), k_t^{ctr}(\mathbf{x}, \mathbf{x}'))$ are completely determined by the parameters $\mathcal{P}_t^i$, $m_t^i(\mathbf{p})$, $\zeta_t^i(\mathbf{p})$ and $\mathcal{P}_t^{ctr}$, $m_t^{ctr}(\mathbf{p})$, $\zeta_t^{ctr}(\mathbf{p})$, respectively, it is sufficient to show that $|m_t^i(\mathbf{p}) - m_t^{ctr}(\mathbf{p})| \to 0$ and $|\zeta_t^i(\mathbf{p}) - \zeta_t^{ctr}(\mathbf{p})| \to 0$ for all $i \in \mathcal{V}$, $\mathbf{p} \in \mathcal{P}_\#$. Let $\mathbf{p} \in \mathcal{P}_\#$ be arbitrary and note that $m_0^i(\mathbf{p}) = m_0^{ctr}(\mathbf{p}) = 0$ and $\zeta_0^i(\mathbf{p}) = \zeta_0^{ctr}(\mathbf{p}) = 0$ since no pseudo points have been observed initially. Expand (31) recursively to obtain $m_t^{ctr}(\mathbf{p})$

and $\zeta_t^{ctr}(\mathbf{p})$ in terms of the observation statistics:

$$m_t^{ctr}(\mathbf{p}) = \sum_{\tau=0}^{t} \sum_{i=1}^{n} \pi_i \tilde{m}_\tau^i(\mathbf{p}),$$

$$\zeta_t^{ctr}(\mathbf{p}) = \frac{1}{m_t^{ctr}(\mathbf{p})} \sum_{\tau=0}^{t} \sum_{i=1}^{n} \pi_i \tilde{m}_\tau^i(\mathbf{p})\tilde{\zeta}_\tau^i(\mathbf{p}). \tag{32}$$

Similarly, expand (30) to obtain $m_t^i(\mathbf{p})$ and $\zeta_t^i(\mathbf{p})$ in terms of the observation statistics:

$$m_t^i(\mathbf{p}) = \sum_{\tau=0}^{t} \sum_{j=1}^{n} \left[W^{t-\tau}\right]_{ij} \tilde{m}_\tau^j(\mathbf{p}),$$

$$\zeta_t^i(\mathbf{p}) = \frac{1}{m_t^i(\mathbf{p})} \sum_{\tau=0}^{t} \sum_{j=1}^{n} \left[W^{t-\tau}\right]_{ij} \tilde{m}_\tau^j(\mathbf{p})\tilde{\zeta}_\tau^j(\mathbf{p}), \tag{33}$$

where the weights $\left[W^{t-\tau}\right]_{ij}$ appear since the data $\tilde{m}_\tau^j(\mathbf{p})$ and $\tilde{\zeta}_\tau^j(\mathbf{p})$ propagate through the network with weight matrix W and reach robot i via all paths of length $t - \tau$. Alternatively, (33) can be viewed as the solution of the discrete-time linear time-invariant system in (30) with transition matrix $\Phi(t,\tau) = W^{t-\tau}$, $t \geq \tau$. Since the data collection stops at some finite time T, $\tilde{m}_t^i(\mathbf{p}) = \tilde{\zeta}_t^i(\mathbf{p}) = 0$ for all $t > T$, $i \in \mathcal{V}$. The convergence of (33) to (32) is concluded from the fact that $[W^t]_{ij} \to \pi_j > 0$ since W is a row-stochastic nonnegative and primitive matrix. $\square$

Prop. 3 is a similar result to [61, Thm. 3], where it is shown that, if the weight matrix W is doubly stochastic, a distributed parameter estimator is as efficient as any centralized parameter estimator. However, Prop. 3 applies to distributed function estimation using an incrementally growing set of parameters and re-weights the observations used by the centralized estimator via the stationary distribution π of W to ensure convergence even when W is not doubly stochastic.

D. Echoless Distributed GP Regression

The distributed pseudo point update we derived in (30) is not efficient for two reasons. First, convergence to the central GP estimate is guaranteed only in the limit, as $t \to \infty$ (Prop. 3). Second, every time robots exchange messages, all information they have must be sent. This is inefficient as may be seen in the proof of Prop. 3, the observations are exchanged an infinite number of times (echos in the network). To address these limitations, we label the communication messages with the list of robots that have already received them and show that convergence to the centralized estimate can, in fact, be achieved in finite time.

Let $\tilde{\Theta}_t^i := \{\tilde{\mathcal{P}}_t^i, \tilde{m}_t^i(\tilde{\mathcal{P}}_t^i), \tilde{\zeta}_t^i(\tilde{\mathcal{P}}_t^i), \ell_t^i\}$ define a mini-batch of observations for robot i. At time t, $\tilde{\Theta}_t^i$ contains the new observations $\tilde{\mathcal{P}}_t^i, \tilde{m}_t^i(\tilde{\mathcal{P}}_t^i), \tilde{\zeta}_t^i(\tilde{\mathcal{P}}_t^i)$ of robot i as well as a list of robots ℓ_t^i that have already received this mini-batch. The list ℓ_t^i is initialized by $\{i\}$. Additionally, for each robot i, we define a set of mini-batches $\mathcal{B}_{t+1}^i$ that the robot should use at time t to update its GP parameters. The mini-batch set $\mathcal{B}_t^i$ from the previous time step contains old mini-batches that robot i should transmit to its neighbors. Inspired by the similarity of (32) and (33), we propose a distributed protocol which ensures:
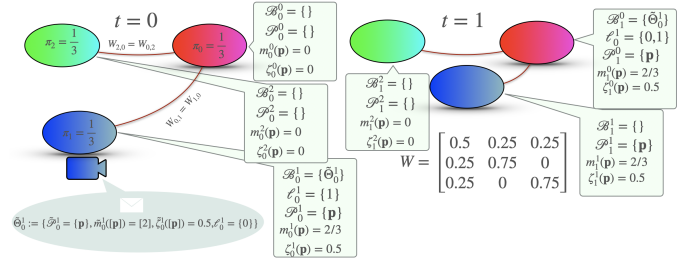

Fig. 4: Visualization of the distributed GP parameter update in (34) in a network with three nodes (red, green, blue). The node parameters are shown at time $t = 0$ and $t = 1$. Only a single observation with one pseudo point is received by node 1 (blue) at time $t = 0$ and is propagated to node 0 (red) at time $t = 1$.

- each mini-batch visits each robot once rather than echoing in the network, relying on ℓ_t^i to keep track of visited robots,
- convergence to the centralized GP distribution is achieved in finite and minimum time by picking the stationary distribution π of W as the coefficient in (33).

The distributed update of the parameters of robot i at time step t is:

$$\mathcal{B}_{t+1}^i = \bigcup_{\tilde{\Theta}_\tau^j \in \mathcal{B}_t^r, r \in \mathcal{N}_i, i \notin \ell_\tau^j} \tilde{\Theta}_\tau^j \cup \tilde{\Theta}_{t+1}^i$$

$$\ell_\tau^j = \ell_\tau^j \cup \{i\} \text{ for all } \tilde{\Theta}_\tau^j \in \mathcal{B}_{t+1}^i$$

$$\mathcal{P}_{t+1}^i = \bigcup_{\tilde{\Theta}_\tau^j \in \mathcal{B}_{t+1}^i} \mathcal{P}_\tau^j \cup \mathcal{P}_t^i$$

$$m_{t+1}^i(\mathbf{p}) = m_t^i(\mathbf{p}) + \sum_{\tilde{\Theta}_\tau^j \in \mathcal{B}_{t+1}^i} \pi_j \tilde{m}_\tau^j(\mathbf{p}) \tag{34}$$

$$\zeta_{t+1}^i(\mathbf{p}) = \frac{m_t^i(\mathbf{p})\zeta_t^i(\mathbf{p}) + \sum_{\tilde{\Theta}_\tau^j \in \mathcal{B}_{t+1}^i} \pi_j \tilde{m}_\tau^j(\mathbf{p})\tilde{\zeta}_\tau^j(\mathbf{p})}{m_{t+1}^i(\mathbf{p})}$$

We prove below that this distributed update rule converges in finite time to the centralized GP distribution. Compared with (30), the distributed update in (34) is able to achieve finite-time convergence because it uses the weights π from the stationary distribution of W right away, instead of processing the same information an infinite number of times to determine the final weights. Moreover, (30) stipulates that two robots should exchange all of their information at each time step, which is very inefficient in practice. The mini-batch messages in (34) allow the robots to exchange only the latest information and guarantee that each observation reaches each agent once. A visualization of this method is shown in Fig. 4.

Proposition 4. *Let $\tilde{\mathcal{D}}_t^i = (\tilde{\mathcal{X}}_t^i, \tilde{\mathbf{y}}_t^i)$ be the data received by robot i at time t, associated with pseudo points $\tilde{\mathcal{P}}_t^i \subset \mathcal{P}_\#$ and number of observations $\tilde{m}_t^i(\mathbf{p})$ and average observation $\tilde{\zeta}_t^i(\mathbf{p})$ for $\mathbf{p} \in \mathcal{P}_\#$. If the data streaming stops at some time $T < \infty$, then at time $t = T + n - 1$, the distributions $\mathcal{GP}(\mu_t^i(\mathbf{x}), k_t^i(\mathbf{x},\mathbf{x}'))$ maintained by each robot i, specified according to (11) with parameters in (34) are exactly equal to the distribution $\mathcal{GP}(\mu_t^{ctr}(\mathbf{x}), k_t^{ctr}(\mathbf{x},\mathbf{x}'))$ of the centralized estimator with parameters in (31), i.e., $\mu_t^i(\mathbf{x}) = \mu_t^{ctr}(\mathbf{x})$ and $k_t^i(\mathbf{x},\mathbf{x}') = k_t^{ctr}(\mathbf{x},\mathbf{x}')$ almost surely for all $i \in \mathcal{V}$, $\mathbf{x}, \mathbf{x}'$.*

Proof. As in the proof of Prop. 3, it is sufficient to show that at $t = T + n - 1$, $m_t^i(\mathbf{p}) = m_t^{ctr}(\mathbf{p})$ and $\zeta_t^i(\mathbf{p}) = \zeta_t^{ctr}(\mathbf{p})$ for all $i \in \mathcal{V}$, $\mathbf{p} \in \mathcal{P}_\#$. As before, we express $m_t^i(\mathbf{p})$ and $\zeta_t^i(\mathbf{p})$ in terms of $\tilde{m}_\tau^j(\mathbf{p})$ and $\tilde{\zeta}_\tau^j(\mathbf{p})$ for arbitrary $\mathbf{p} \in \mathcal{P}_\#$ and $\tau \le t$. The key is to realize whether mini-batch $\tilde{\Theta}_\tau^j$ is received by robot i. Since the mini-batch exchanges are happening based on the communication graph structure, the elements of $W^{t-\tau}$ determine which robots have received a mini-batch released at time τ by time t. Precisely, if $[W^{t-\tau}]_{ij} > 0$, then robot i has received mini-batch $\tilde{\Theta}_\tau^j$ by time t and otherwise, if $[W^{t-\tau}]_{ij} = 0$, it has not received it. Let $\text{sign}(x)$ denote the sign of a scalar x with $\text{sign}(0) = 0$. Expanding (34) recursively leads to:

$$m_t^i(\mathbf{p}) = \sum_{\tau=0}^{t} \sum_{i=1}^{n} \text{sign}([W^{t-\tau}]_{ij}) \pi_j \tilde{m}_\tau^j(\mathbf{p}) \tag{35}$$

$$\zeta_t^i(\mathbf{p}) = \frac{1}{m_t^i(\mathbf{p})} \sum_{\tau=0}^{t} \sum_{j=1}^{n} \text{sign}([W^{t-\tau}]_{ij}) \pi_j \tilde{m}_\tau^j(\mathbf{p}) \tilde{\zeta}_\tau^j(\mathbf{p})$$

Since the data collection stops at some finite time T, $\tilde{m}_\tau^i(\mathbf{p}) = \tilde{\zeta}_\tau^i(\mathbf{p}) = 0$ for all $\tau > T$, $i \in \mathcal{V}$. Comparing (33) and (32), equality of $\mu_t^i(\mathbf{x})$ and $\mu_t^{ctr}(\mathbf{x})$ and $k_t^i(\mathbf{x}, \mathbf{x}')$ and $k_t^{ctr}(\mathbf{x}, \mathbf{x}')$ at $t = T + n - 1$ is concluded by the fact that $[W^{n-1}]_{ij} > 0$ because the network is connected. $\square$

VII. Distributed Metric-Semantic Mapping

We apply the distributed GP regression technique developed in Sec. VI to the multi-robot metric-semantic TSDF mapping problem. Each robot i receives local distance and class observations $\{\lambda_{t+1,k}^i, c_{t+1,k}^i\}$, which are transformed using the procedure in Sec. V-A into training data sets $\tilde{\mathcal{D}}_{t+1,l}^i = (\tilde{\mathcal{X}}_{t+1,l}^i, \tilde{\mathbf{y}}_{t+1,l}^i)$ for estimating the TSDFs $\{f_l(\mathbf{x})\}$ of the different object classes. Each dataset $\tilde{\mathcal{D}}_{t+1,l}^i$ is compressed into a set of pseudo points $\tilde{\mathcal{P}}_{t+1,l}^i$ with associated number of observations $\tilde{m}_{t+1,l}^i(\mathbf{p})$ and average observation $\tilde{\zeta}_{t+1,l}^i(\mathbf{p})$ for $\mathbf{p} \in \tilde{\mathcal{P}}_{t+1,l}^i$. Each robot maintains a separate GP $\mathcal{GP}(\mu_{t,l}^i(\mathbf{x}), k_{t,l}^i(\mathbf{x}, \mathbf{x}'))$ for each class TSDF $f_l(\mathbf{x})$. In the multi-robot case, the GP distributions of robot i are updated simultaneously and independently for all classes using the new class-specific observation data $\tilde{\mathcal{P}}_{t+1,l}^i$, $\tilde{m}_{t+1,l}^i(\tilde{\mathcal{P}}_{t+1,l}^i)$, $\tilde{\zeta}_{t+1,l}^i(\tilde{\mathcal{P}}_{t+1,l}^i)$ as well as information from the neighboring robots in the form of class-specific mini-batches $\mathcal{B}_{t+1,l}^i$ as described in (34). To make the GP models scalable to large environments, we organize the pseudo points $\mathcal{P}_{t,l}^i$ for each robot i and class l in an octree data structure, as in Sec. V-D, and predict the class of a query point via the method in Sec. V-C. Prop. 4 guarantees that the local TSDF GPs at each robot converge to a common GP, which is equivalent to the one that would be obtained by centralized sparse GP regression. Moreover, when the streaming of new observations stops, the convergence happens in finite time as soon as each observation is received by each robot exactly once. In other words, there is no unnecessary communication in the form of information exchange echo in the network.

VIII. Evaluation using 2-D Simulated Data

In this section, we evaluate our semantic TSDF mapping approach in 2-D simulated environments. We first demonstrate
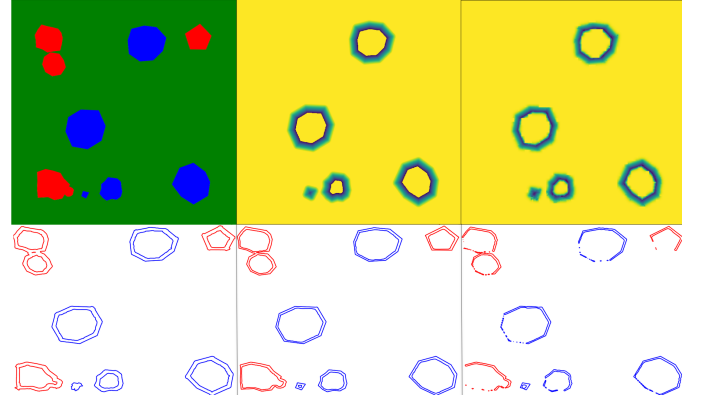

Fig. 5: Ground-truth 2-D simulated environment (top left) with two object classes (red, blue), ground-truth TSDF for the blue class (top middle), and reconstructed TSDF with $frame\ size = 10$ (top right). The reconstructed TSDF boundaries are shown for three different $frame\ size$ parameters on the bottom row: 10 (bottom left), 3 (bottom middle), 2 (bottom right). Sharp edges are captured better with $frame\ size$ 3 vs. 10 but using $frame\ size$ less that 3 caused missing parts at the boundaries.

the qualitative and quantitative performance of the single-robot approach of Sec. V. Then, we report results for the multi-robot approach of Sec. VII using three robots to map the same environment collaboratively. In all experiments, we employ a sparse Matérn kernel ($\nu = 3/2$) [21]. We choose the workspace discretization $\mathcal{P}_\#$ as a grid with resolution $voxel\ size$. Given a training point $\hat{\mathbf{x}}$ in (16), we choose a square region of pseudo points from $\mathcal{P}_\#$ around $\hat{\mathbf{x}}$. These pseudo points are used to construct the training set in (17) around the sensor hit points $\hat{\mathbf{x}}$, instead of a circle with radius ϵ. We call the number of pseudo points on the edge of the square region $frame\ size$, and choose it so that $(frame\ size - 1) \times voxel\ size \ge 2\epsilon$.

A. Single-Robot 2-D Evaluation

We generate random 2-D environments (see Fig. 5) and robot trajectories by sampling poses sequentially and keeping the ones that are in free space. Observations are obtained along the robot trajectories using a simulated distance-class sensor. We apply our incremental sparse GP regression method to obtain a probabilistic TSDF map and compare it with the ground truth TSDF.

1) TSDF Accuracy: One sample environment from our 2-D simulation with the ground truth and reconstructed TSDF and boundaries is shown in Fig. 5. Our method provides continuous probabilistic TSDF estimates. The choice of $frame\ size$ is very dependent on the desired truncation value for the SDF reconstruction. Larger $frame\ size$ allows estimating larger truncation values but incurs additional computation cost. The precision and resilience to measurement noise of our method are evaluated in Fig. 6. The test points are chosen from a grid with resolution $0.5 \times voxel\ size$ within the truncation distance from the ground-truth object boundaries.

2) Classification Accuracy: We evaluate the average precision and recall of our posterior classification over 50 random 2-D maps. In each map, we pick uniformly distributed random
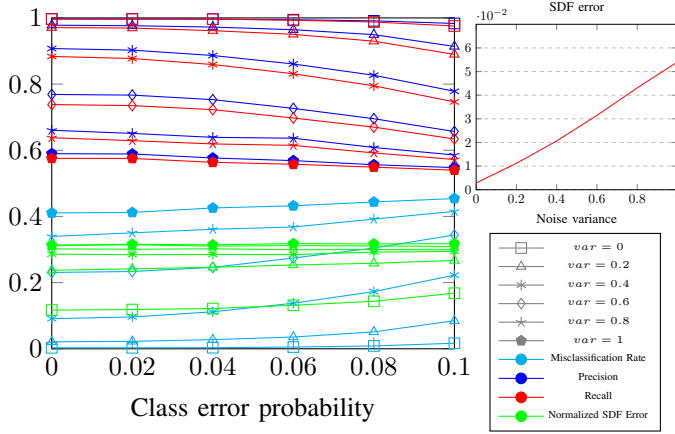
Fig. 6: Misclassification Rate, Precision, Recall, and Normalized SDF Error for different class error probability and distance noise variance. The top right plot shows the average SDF error over 10 random maps with a 100 random observations each, with $voxel\ size = 0.1$, $max(N) = 100$, $\delta = 1.2$.
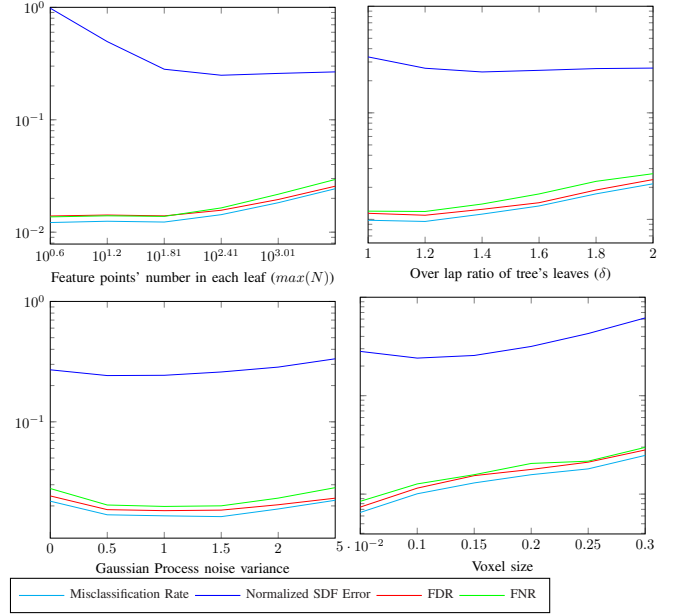

Fig. 7: Misclassification rate, normalized SDF error, False Discovery Rate (FDR), and False Negative Rate (FNR) as a function of the number of pseudo points per octree support region $(max(N))$, support region overlap ratio (δ), GP noise variance σ^2, and workspace discretization $(voxel\ size)$. The default parameter values are $\delta = 1.5$, $max(N) = 100$, $\sigma^2 = 1$, $voxel\ size = 0.1$. Class and distance measurements with class error probability of 0.05 and distance noise variance 0.5 are obtained from 100 random observations in each of 50 random 2-D maps. Test points are selected within a threshold of 0.05 from the ground truth class boundaries.

points along the obstacle boundaries, and calculate the SDF error and the class-detection accuracy. Since the values are symmetric for binary classification, we present the average precision and recall over the two classes in Fig. 6. The figure shows that the misclassification rate, precision, recall, and SDF error are not very sensitive to class error probability. The misclassification rate is the ratio of all to the misclassified test points. The SDF error is the average absolute value difference between the estimated and ground-truth SDF values. We report normalized SDF error: $\frac{\text{SDF error}}{voxel\ size}$. Fig. 7 investigates the effect of the parameters of our algorithm on misclassification rate, normalized SDF error, False Discovery Rate (FDR := 1 − Precision), and False Negative Rate (FNR := 1 − Recall). We see that the misclassification rate, FNR, and FDR respond similarly to parameter variations.

Increasing the maximum number of pseudo points per octree support region, $max(N)$, improves the (normalized) SDF error. The improvement is significant at first but after a certain octree support region size, even exponential increases in $max(N)$ do not significantly affect the SDF error. The classification measures improve slightly with an initial increase in $max(N)$. Increasing δ has a similar effect on all the performance measures. Increasing the GP noise variance σ^2 at first improves all the measures but then it worsens them. An incorrect choice of σ^2 is critical to the method, but affects the misclassification rate smoothly so, it must be in the right region, but as long as the value of σ^2 is in the right ballpark, choosing the optimal σ^2 is not critical.

B. Multi-Robot 2-D Evaluation

Next, we evaluate the distributed GP regression in a three-robot simulation and investigate the convergence of the local GP estimates of each robot to a centralized GP estimate. We use the same random polygonal 2-D environments with two object classes but this time generate trajectories for three different robots (see Fig. 8). The robots communicate with

each other over a graph with a fixed weight matrix:

$$W = \begin{bmatrix} 0.5 & 0.25 & 0.25 \\ 0.25 & 0.75 & 0 \\ 0.25 & 0 & 0.75 \end{bmatrix}. \tag{36}$$

The GP regression parameters at each robot are the same as the defaults in Sec. VIII-A.

To verify Prop. 4 empirically, we compare the mean absolute error (MAE) between the GP prediction of an individual robot i and the centralized estimator ctr using all robot observations as described in Sec. VI-C. Specifically, at each time step t, we consider all classes l and associated pseudo points $\mathcal{P}_{t,l}^{ctr}$ that have been observed by the centralized estimator and calculate the mean MAE as:

$$MAE_t = \frac{1}{L_t |\mathcal{P}_{t,l}^{ctr}|} \sum_{\ell} \sum_{\mathbf{p} \in \mathcal{P}_{t,l}^{ctr}} \left| \mu_{t,l}^i(\mathbf{p}) - \mu_{t,l}^{ctr}(\mathbf{p}) \right|, \tag{37}$$

where L_t is the number of observed object classes by time t. The variance MAE is computed equivalently to (37) with $\mu_{t,l}^i(\mathbf{p})$ and $\mu_{t,l}^{ctr}(\mathbf{p})$ replaced by $k_{t,l}^i(\mathbf{p}, \mathbf{p})$ and $k_{t,l}^{ctr}(\mathbf{p}, \mathbf{p})$.

Fig. 8 shows the final reconstructions of one robot and the centralized estimator. As expected, the final reconstructions are identical and convergence happens in finite time. The behavior of the mean and variance MAE curves is similar. This is expected because the distance between the local and centralized GP parameters is due to unobserved information rather than stochastic noise. We see that the MAE curves approach 0 quickly. Several peaks are observed in the curves

Fig. 9: Single-robot reconstruction of the Cow and Lady dataset [37]. Red hues indicate lower TSDF variance.
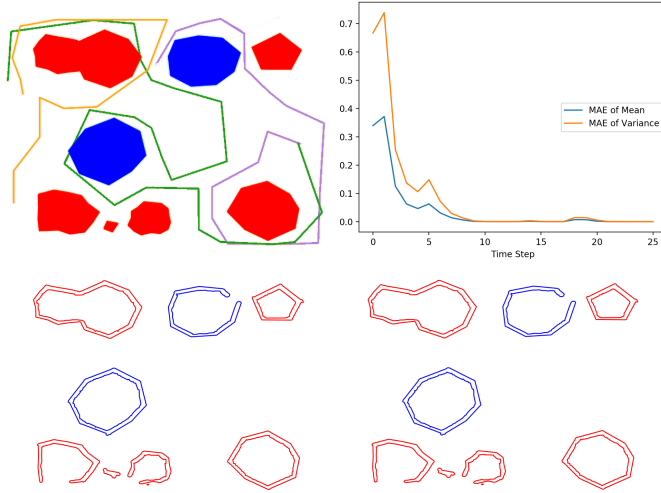
Fig. 8: Three robot trajectories (green, orange, purple) in a 2-D simulated environment (top left) with two object classes (red, blue). The zero level-sets of the TSDF reconstructions for the two classes by centralized GP regression (bottom left) and distributed GP regression from the perspective of the orange robot (bottom right) are shown. As expected, due to Prop. 2, the centralized and individual robot reconstructions are identical. This is verified quantitatively in the GP mean and variance mean absolute error (MAE) plot (top right). The initial GP parameters for each robot and object class were $\mu_{0,l}^i(\mathbf{x}) = 0.5, k_{0,l}^i(\mathbf{x}, \mathbf{x}) = 1$.

when new sections of the environment that are not visible to robot i are observed by another robot in the network. The new information disseminates in the network and the MAE curves approach zero again.

IX. Evaluation using 3-D Real Data

In this section, we evaluate our semantic TSDF mapping approach using real RGB-D data from physical 3-D environments. We demonstrate the qualitative and quantitative performance of the single-robot approach of Sec. V and the multi-robot approach of Sec. VII, using three robots to map the same environment collaboratively. As in the 2-D experiments in Sec. VIII, we use a sparse Matérn kernel ($v = 3/2$) and a grid of potential pseudo points $\mathcal{P}_\#$ with resolution $voxel\ size$. Given a query point $\hat{\mathbf{x}}$, we choose a cubic region around it such that $(frame\ size - 1) \times voxel\ size \geq 2 \times \epsilon$ to construct the training data in (16). All points from $\mathcal{P}_\#$ that lie in the cubic region are chosen as pseudo points associated with $\hat{\mathbf{x}}$.

A. Single-Robot 3-D Evaluation

We compare our method to the incremental Euclidean signed distance mapping method Fiesta [38] on the Cow and Lady dataset [37]. We also demonstrate the 3-D semantic reconstruction performance of our method on the SceneNN dataset [72].

1) Cow and Lady Dataset: The reconstruction of the Cow and Lady dataset with 829 depth images and known camera trajectory by the single-robot TSDF GP regression of Sec. V is shown in Fig. 9. A triangular mesh is extracted from the mean TSDF prediction using the marching cubes algorithm [73].

The reconstruction time and error with respect to the ground-truth scene point cloud provided by the dataset are reported in Fig. 10. The error of Fiesta with default parameters is shown as well. Similar to the 2-D simulations, increasing the maximum number of pseudo points $max(N)$ per octree support region improves the SDF error of our approach. The improvement is significant at first and less pronounced afterwards. Conversely, the computation time decreases at first because the number of leaves in the octree decreases and then increases afterwards as the GP covariance matrices get larger. Increasing δ leads to an insignificant improvement in the SDF error at the expense of a significant reconstruction time increase. Increasing the GP noise variance improves the SDF error at first (especially when the error is close to zero) but worsens is afterwards without significant impact on time. As $voxel\ size$ varies, our method outperforms Fiesta noticeably.

2) SceneNN Dataset: We evaluate the classification accuracy of our method on the SceneNN dataset in Fig. 11. The GP posterior is evaluated on a test grid with resolution $0.5 \times voxel\ size$. The test points with posterior variance less than a threshold are used to reconstruct a triangular mesh via the marching cubes algorithm [73]. We use Prop. 2 for classification. The effect of the different parameters on the performance is illustrated in Fig. 11. Increasing $max(N)$ improves both classification and TSDF reconstruction results. The improvement after $max(N) = 100$ is negligible but time increases significantly. Increasing δ improves the TSDF reconstruction significantly at first. After $\delta = 1.4$, the improvement is negligible. As seen in the 2-D simulations, choosing a correct magnitude for the GP noise variance σ^2 is very important for both the classification and TSDF reconstruction but choosing the optimal value for σ^2 is not critical.

B. Multi-Robot 3-D Evaluation

Finally, we evaluate our distributed GP regression on the Cow and Lady and SceneNN datasets. To imitate data collection by multiple robots, we split the RGB-D image sequences into equal parts and consider each as data obtained by a different robot. As in the 2-D simulation, we use three robots with communication structure specified by the weight matrix
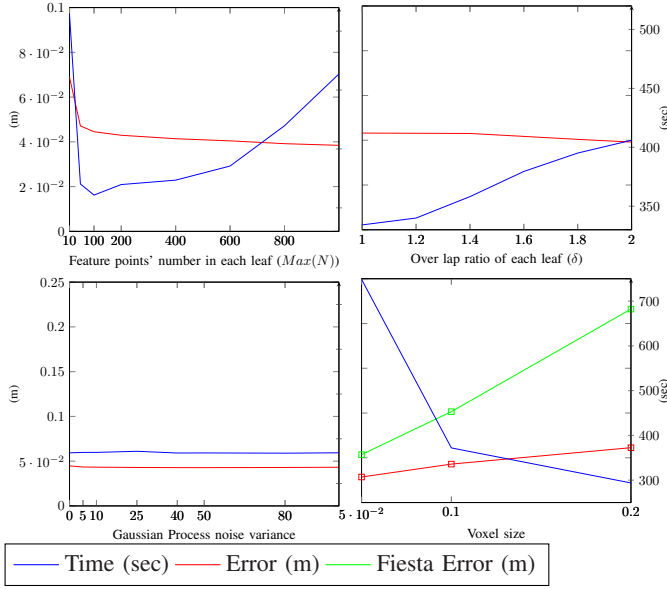
Fig. 10: Evaluation of the SDF reconstruction time (sec) and error (m) of our incremental sparse GP regression algorithm on the Cow and Lady dataset [37] and in comparison with Fiesta [38]. The errors are evaluated with respect to the ground-truth scene point cloud provided by the dataset. Training is done with 829 depth images and known camera trajectory. The default parameters for our algorithm are $max(N) = 200$, $\delta = 1.5$, $\sigma^2 = 25$, $voxel\ size = 0.1$, $frame\ size = 5$, and SDF truncation value $3 \times voxel\ size$.

X. CONCLUSION

This paper developed a Bayesian inference method for online probabilistic metric-semantic mapping via scalable Gaussian Processes regression of semantic class signed distance functions. Our algorithm was enabled by several key ideas. First, repeated observations of the same environment locations can be compressed before training a GP regression method without any effect on the posterior distribution. This, combined with an overlapping-leaf octree data structure of pseudo points, allowed the development of an incremental sparse GP regression technique, which scales to large domains. Second, instead of explicit modeling of class likelihoods and reliance on computationally challenging GP classification techniques, the presence of distance measurements allows independent GP regression for each class. A class probability mass function can still be recovered at test time based on the distance distributions, and its accuracy was shown empirically to be resilient to increasing classification error rates. Third, distributed parameter estimation techniques based on consensus can be extended to distributed function estimation by relying on incrementally growing pseudo points. This enables distributed incremental sparse GP regression, guaranteed to converge in finite-time to the same distribution as that of a centralized estimator without relying on multi-hop communication. Our method enables robot teams to collaboratively build dense metric-semantic maps of unknown environments using streaming RGB-D measurements. This offers a promising direction for future research in semantic task specifications and uncertainty-aware task planning.

REFERENCES

[1] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Conference on Computer Graphics and Interactive Techniques*, 1996, pp. 303–312.

[2] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Eurographics Symposium on Geometry Processing*, 2006.

[3] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, 2013.

[4] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 6, pp. 1–11, 2013.

[5] 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.

[6] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," in *2017 IEEE International Conference on Robotics and automation (ICRA)*. IEEE, 2017, pp. 4628–4635.

[7] A. Hermans, G. Floros, and B. Leibe, "Dense 3D Semantic Mapping of Indoor Scenes from RGB-D Images," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 2631–2638.

[8] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg, "Joint semantic segmentation and 3d reconstruction from monocular video," in *European Conference on Computer Vision*. Springer, 2014, pp. 703–718.

[9] J. Hensman, A. Matthews, and Z. Ghahramani, "Scalable Variational Gaussian Process Classification," in *International Conference on Artificial Intelligence and Statistics*, 2015, pp. 351–360.

[10] T. Galy-Fajou, F. Wenzel, C. Donner, and M. Opper, "Multi-class gaussian process classification made conjugate: Efficient inference via data augmentation," in *Uncertainty in Artificial Intelligence Conference*, 2020, pp. 755–765.

[11] M. Ghaffari Jadidi, L. Gan, S. Parkison, J. Li, and R. Eustice, "Gaussian Processes Semantic Map Representation," *arXiv:1707.01532*, 2017.

[12] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Springer, 2003, pp. 63–71.

W in (36). Each robot uses the distributed update rule in (34) and communication continues for 2 rounds after the last RGB-D image from the individual robot sequences is received. The parameters of the individual robots are the same as in the single-robot experiments in Sec. IX-A. The choice of additional rounds is due to Prop. 4, where we showed theoretically that $T + n - 1$ rounds are needed, where T is the observation sequence length and n is the number of robots, for the local GP distributions to agree with that of a centralized GP estimator. As in the 2-D simulations, to verify Prop. 4 empirically, we compare the mean absolute error (MAE) in (37) between the GP mean and variance of an individual robot and the centralized estimator.

The results from the Cow and Lady dataset are reported in Fig. 12 and Fig. 13, while those from the SceneNN dataset are reported in Fig. 14 and Fig. 15. The local and centralized reconstruction results are identical in both data sets, which confirms the expected theoretical consistency. The mean and variance MAE curves also behave similarly in both data sets because the errors in the local GP regression are due to unobserved information, that has not yet been received by the robot, rather than measurement noise. As in the 2-D simulation, the peaks in the MAE curves are due to another robot in the network observing a new region that has not yet been observed by this robot. These peaks quickly decrease, which indicates the fast empirical convergence of the distributed sparse GP algorithm.
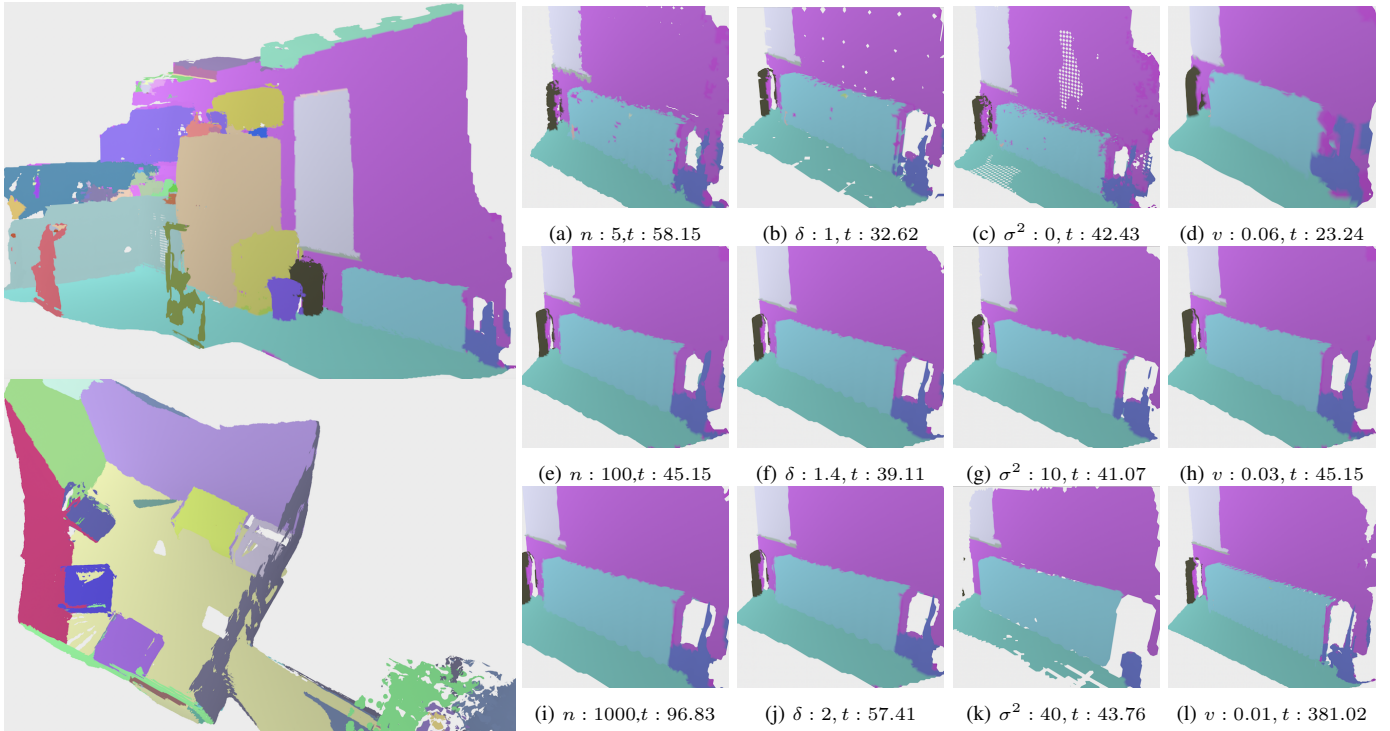
Fig. 11: Single-robot reconstructions of sequence 255 (top left), containing 2450 RGB-D images and 85 semantic categories (in random colors), and sequence 011 (bottom left), containing 3700 RGB-D images and 61 semantic categories (in random colors), of the SceneNN dataset [72]. The incremental sparse GP TSDF mapping process took 1040.41 sec. for sequence 255 and 1885.72 sec. for sequence 011. The following default parameters were used for the octree: $\delta = 1.5$, $n = max(N) = 100$ and the GP training: $\sigma^2 = 3$, $v = voxel\ size = 0.03$, $frame\ size = 1$. On the right we see the effect of these parameters (t is time in seconds) on the metric-semantic reconstruction over 140 RGB-D images.

Sub-captions:
- (a) $n : 5, t : 58.15$
- (b) $\delta : 1, t : 32.62$
- (c) $\sigma^2 : 0, t : 42.43$
- (d) $v : 0.06, t : 23.24$
- (e) $n : 100, t : 45.15$
- (f) $\delta : 1.4, t : 39.11$
- (g) $\sigma^2 : 10, t : 41.07$
- (h) $v : 0.03, t : 45.15$
- (i) $n : 1000, t : 96.83$
- (j) $\delta : 2, t : 57.41$
- (k) $\sigma^2 : 40, t : 43.76$
- (l) $v : 0.01, t : 381.02$

[13] S. O'Callaghan and F. Ramos, "Gaussian process occupancy maps," *The International Journal of Robotics Research (IJRR)*, vol. 31, no. 1, pp. 42–62, 2012.

[14] S. Kim and J. Kim, "Occupancy Mapping and Surface Reconstruction Using Local Gaussian Processes With Kinect Sensors," *IEEE Trans. on Cybernetics*, vol. 43, no. 5, pp. 1335–1346, 2013.

[15] M. G. Jadidi, J. V. Miró, R. Valencia, and J. Andrade-Cetto, "Exploration on Continuous Gaussian Process Frontier Maps," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014, pp. 6077–6082.

[16] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," in *Advances in neural information processing systems*, 2006, pp. 1257–1264.

[17] J. Hensman, N. Durrande, and A. Solin, "Variational Fourier features for Gaussian processes," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 5537–5588, 2017.

[18] A. Koppel, "Consistent online Gaussian Process regression without the sample complexity bottleneck," in *American Control Conference (ACC)*, 2019, pp. 3512–3518.

[19] A. Koppel, A. S. Bedi, K. Rajawat, and B. M. Sadler, "Optimally compressed nonparametric online learning," *IEEE Signal Processing Magazine*, 2020.

[20] V. Tresp, "A bayesian committee machine," *Neural computation*, vol. 12, no. 11, pp. 2719–2741, 2000.

[21] S. Kim and J. Kim, "Recursive Bayesian Updates for Occupancy Mapping and Surface Reconstruction," in *Australasian Conference on Robotics and Automation (ACRA)*, 2014.

[22] M. Bauer, M. van der Wilk, and C. E. Rasmussen, "Understanding probabilistic sparse gaussian process approximations," in *Advances in neural information processing systems*, 2016, pp. 1533–1541.

[23] C. E. Rasmussen and Z. Ghahramani, "Infinite mixtures of gaussian process experts," in *Advances in neural information processing systems*, 2002, pp. 881–888.

[24] A. Nedić, A. Olshevsky, and C. A. Uribe, "Distributed learning for cooperative inference," *arXiv preprint:1704.02718*, 2017.

[25] A. Jadbabaie, P. Molavi, A. Sandroni, and A. Tahbaz-Salehi, "Non-bayesian social learning," *Games and Economic Behavior*, vol. 76, no. 1, pp. 210–225, 2012.

[26] E. Zobeidi, A. Koppel, and N. Atanasov, "Dense incremental metric-semantic mapping via sparse gaussian process regression," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[27] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[28] E. Vespa, N. Nikolov, M. Grimm, L. Nardi, P. H. J. Kelly, and S. Leutenegger, "Efficient Octree-Based Volumetric SLAM Supporting Signed-Distance and Occupancy Mapping," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1144–1151, 2018.

[29] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conf. on Computer Vision*, 2014.

[30] R. Dubé, A. Cramariuc, D. Dugas, H. Sommer, M. Dymczyk, J. Nieto, R. Siegwart, and C. Cadena, "SegMap: Segment-based mapping and localization using data-driven descriptors," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 339–355, 2020.

[31] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.

[32] J. Behley and C. Stachniss, "Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments," in *Robotics: Science and Systems*, 2018.

[33] L. Teixeira and M. Chli, "Real-time mesh-based scene estimation for aerial inspection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4863–4869.

[34] E. Piazza, A. Romanoni, and M. Matteucci, "Real-time cpu-based large-scale three-dimensional mesh reconstruction," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1584–1591, 2018.

[35] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-Time Dense Surface Mapping and Tracking," in *IEEE Int. Symposium on Mixed and Augmented Reality*, 2011, pp. 127–136.

[36] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and
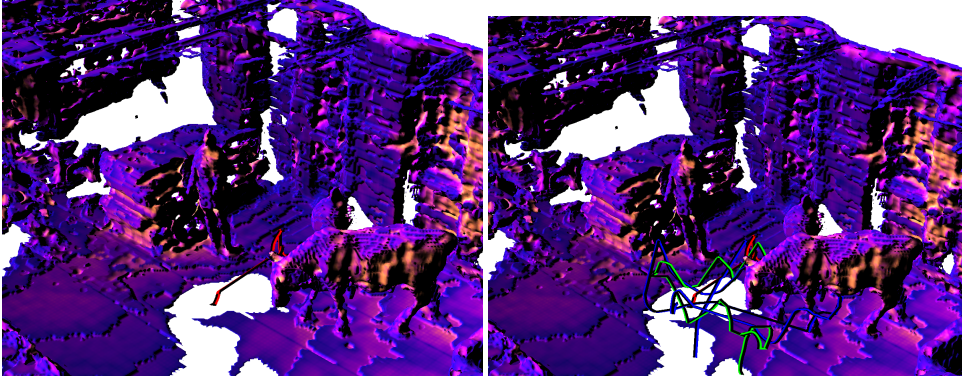
Fig. 12: The Cow and Lady dataset [37] is divided into three equal sequences of about 275 depth images, and each is considered data obtained by one robot. The three camera trajectories are shown in red, green, and blue on the right. The left plot shows the final reconstruction obtained by the first robot. The right plot shows the final reconstruction obtained by centralized GP regression using the observations of all three robots. The orange hues indicate larger variance. As expected, due to Prop. 4, the reconstruction of robot one is identical with that of the centralized estimator. The initial GP parameters for each robot and object class were $\mu_{0,l}^i(\mathbf{x}) = 0.15$ and $k_{0,l}^i(\mathbf{x}, \mathbf{x}) = 5$.
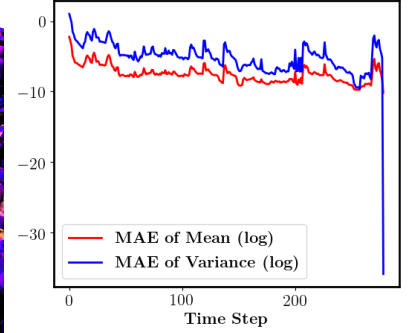
Fig. 13: Log-space plot of the mean absolute error (MAE) between the mean (red) and variance (blue) predictions of robot 1 and centralized GP regression for the sequence in Fig. 12. When the data streaming stops at the end, the MAE approaches zero ($-\infty$ in log space).
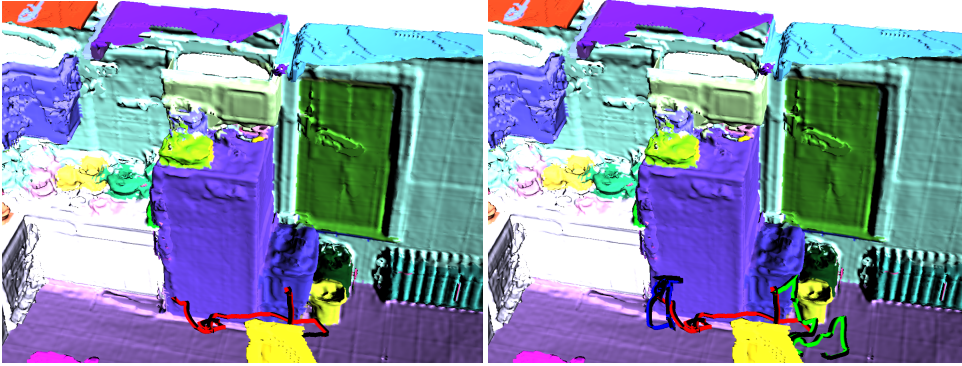

Fig. 14: Sequence 255 of the SceneNN dataset [72] is divided into three equal sequences of about 800 RGB-D images, and each is considered data obtained by a different robot. The three camera trajectories are shown in red, green, and blue on the right. The left plot shows the final metric-semantic reconstruction obtained by the first robot. The right plot shows the final reconstruction obtained by centralized GP regression using the observations of all three robots. As expected, due to Prop. 4, the reconstruction of robot one is identical with that of the centralized estimator. The initial GP parameters for each robot and object class were $\mu_{0,l}^i(\mathbf{x}) = 0.09$ and $k_{0,l}^i(\mathbf{x}, \mathbf{x}) = 5$.
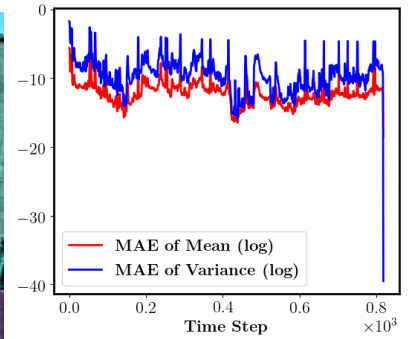
Fig. 15: Log-space plot of the mean absolute error (MAE) between the mean (red) and variance (blue) predictions of robot 1 and centralized GP regression for the sequence in Fig. 14. When the data streaming stops at the end, the MAE approaches zero ($-\infty$ in log space).

S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016.

[37] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2017.

[38] L. Han, F. Gao, B. Zhou, and S. Shen, "Fiesta: Fast incremental euclidean distance fields for online motion planning of aerial robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2019.

[39] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. McDonald, "Real-time large-scale dense RGB-D SLAM with volumetric fusion," *The International Journal of Robotics Research (IJRR)*, vol. 34, no. 4-5, pp. 598–626, 2015.

[40] M. Klingensmith, I. Dryanovski, S. S. Srinivasa, and J. Xiao, "Chisel: Real Time Large Scale 3D Reconstruction Onboard a Mobile Device using Spatially Hashed Signed Distance Fields," in *Robotics: science and systems*, vol. 4. Citeseer, 2015, p. 1.

[41] L. Han and L. Fang, "FlashFusion: Real-time Globally Consistent Dense 3D Reconstruction using CPU Computing," in *Robotics: Science and Systems (RSS)*, 2018.

[42] O. Kähler, V. A. Prisacariu, and D. W. Murray, "Real-time large-scale dense 3d reconstruction with loop closure," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 500–516.

[43] V. Reijgwart, A. Millane, H. Oleynikova, R. Siegwart, C. Cadena, and

J. Nieto, "Voxgraph: Globally Consistent, Volumetric Mapping Using Signed Distance Function Submaps," *IEEE Robotics and Automation Letters*, 2020.

[44] 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*, vol. 4, no. 3, pp. 3037–3044, 2019.

[45] E. Bylow, J. Sturm, C. Kerl, F. Kahl, and D. Cremers, "Real-time camera tracking and 3d reconstruction using signed distance functions." in *Robotics: Science and Systems*, 2013.

[46] H. Oleynikova, M. Burri, Z. Taylor, J. I. Nieto, R. Siegwart, and E. Galceran, "Continuous-time trajectory optimization for online uav replanning," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[47] K. Saulnier, N. Atanasov, G. Pappas, and V. Kumar, "Information theoretic active exploration in signed distance fields," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2020.

[48] S. Kim and J. Kim, "GPmap: A Unified Framework for Robotic Mapping Based on Sparse Gaussian Processes," in *International Conference on Field and Service Robotics*, 2015.

[49] J. Wang and B. Englot, "Fast, accurate gaussian process occupancy maps via test-data octrees and nested bayesian fusion," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2016, pp. 1003–1010.

[50] F. Ramos and L. Ott, "Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent," *The International Journal of*

Robotics Research, vol. 35, no. 14, pp. 1717–1730, 2016.

[51] R. Senanayake and F. Ramos, "Bayesian Hilbert Maps for Continuous Occupancy Mapping in Dynamic Environments," in *Conference on Robot Learning (CoRL)*, ser. Proceedings of Machine Learning Research, vol. 78, 2017, pp. 458–471.

[52] V. Guizilini and F. Ramos, "Learning to Reconstruct 3D Structures for Occupancy Mapping," in *Robotics: Science and Systems*, 2017.

[53] S. Guo and N. A. Atanasov, "Information filter occupancy mapping using decomposable radial kernels," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 7887–7894.

[54] V. Vineet, O. Miksik, M. Lidegaard, M. Nießner, S. Golodetz, V. A. Prisacariu, O. Kähler, D. W. Murray, S. Izadi, P. Pérez, and P. H. S. Torr, "Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 75–82.

[55] S. Sengupta and P. Sturgess, "Semantic octree: Unifying recognition, reconstruction and representation via an octree constrained higher order mrf," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1874–1879.

[56] S. Yang, Y. Huang, and S. Scherer, "Semantic 3D occupancy mapping through efficient high-order CRFs," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 590–597.

[57] Z. Zhao and X. Chen, "Building 3D semantic maps for mobile robots using RGB-D camera," *Intelligent Service Robotics*, vol. 9, no. 4, pp. 297–309, 2016.

[58] K. Zheng and A. Pronobis, "From pixels to buildings: End-to-end probabilistic deep networks for large-scale semantic mapping," *arXiv preprint arXiv:1812.11866*, 2018.

[59] L. Gan, R. Zhang, J. W. Grizzle, R. M. Eustice, and M. Ghaffari Jadidi, "Bayesian spatial kernel smoothing for scalable dense semantic mapping," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 790–797, 2020.

[60] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[61] K. Rahnama Rad and A. Tahbaz-Salehi, "Distributed parameter estimation in networks," in *IEEE Conference on Decision and Control (CDC)*, 2010, pp. 5050–5055.

[62] N. Atanasov, R. Tron, V. M. Preciado, and G. J. Pappas, "Joint estimation and localization in sensor networks," in *IEEE Conference on Decision and Control (CDC)*, 2014, pp. 6875–6882.

[63] A. Nedić, A. Olshevsky, and C. A. Uribe, "Distributed Gaussian learning over time-varying directed graphs," in *Asilomar Conference on Signals, Systems and Computers*, 2016, pp. 1710–1714.

[64] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, "Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models," *The International Journal of Robotics Research*, vol. 36, no. 12, pp. 1286–1311, 2017.

[65] P. Koch, S. May, M. Schmidpeter, M. Kühn, C. Pfitzner, C. Merkl, R. Koch, M. Fees, J. Martin, D. Ammon, and A. Nüchter, "Multi-robot localization and mapping based on signed distance functions," *Journal of Intelligent & Robotic Systems*, vol. 83, no. 3-4, pp. 409–428, 2016.

[66] P. Lajoie, B. Ramtoula, Y. Chang, L. Carlone, and G. Beltrame, "DOOR-SLAM: Distributed, Online, and Outlier Resilient SLAM for Robotic Teams," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1656–1663, 2020.

[67] A. Milioto and C. Stachniss, "Bonnet: An Open-Source Training and Deployment Framework for Semantic Segmentation in Robotics using CNNs," in *IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[68] A. Tahbaz-Salehi and A. Jadbabaie, "A Necessary and Sufficient Condition for Consensus Over Random Networks," *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 791–795, 2008.

[69] A. Nedic and A. Ozdaglar, "Distributed Subgradient Methods for Multi-Agent Optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[70] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, 2005.

[71] F. Saadatniaki, R. Xin, and U. A. Khan, "Decentralized optimization over time-varying directed graphs with row and column-stochastic matrices," *IEEE Transactions on Automatic Control*, vol. 65, no. 11, pp. 4769–4780, 2020.

[72] B.-S. Hua, Q.-H. Pham, D. T. Nguyen, M.-K. Tran, L.-F. Yu, and S.-K. Yeung, "Scenenn: A scene meshes dataset with annotations," in *International Conference on 3D Vision (3DV)*, 2016.

[73] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.