Multi-Robot Adaptive Sampling based on Mixture of Experts Approach to Modeling Non-Stationary Spatial Fields

Kizito Masaba and Alberto Quattrini Li

Abstract—This paper presents an adaptive sampling strategy for a team of robots to model non-stationary spatial fields—i.e., fields with uneven variations—in large environments, with a desired predictive accuracy. Modeling non-stationary heterogeneous fields is essential for many applications, like monitoring air quality or contamination level in lakes. Mainstream adaptive sampling strategies assume stationarity of the environmental phenomenon and use a single model to explain such fields, resulting in inaccurate characterization of unique localized variations. In this paper, we model a non-stationary field as a collection of (infinite, in theory) non-overlapping layers of stationary homogeneous subfields. This approach allows for modeling non-stationary fields using a mixture of experts, where each expert is assigned a particular homogeneous subregion to map. This approach decomposes the environment into smaller homogeneous regions, which allows real-time modeling of large environments. We design a data-driven approach to adaptively identify each stationary layer and define its relationship to other layers. We model the relationship between various sub-regions as a network of experts—the rapidly-sampling adaptive graph. Each robot incrementally builds its own sub-network of experts, which is used to determine where to sample. Several experiments in realistic simulation demonstrate competitive accuracy and sampling efficiency compared to other state-of-the-art methods.

I. INTRODUCTION

We tackle the problem of multi-robot adaptive sampling for large data collection and modeling of a non-stationary spatial field with predefined predictive accuracy. In contrast to stationary fields that exhibit a uniform variation across the environment, non-stationary fields have uneven variations—a property that is more representative of real world phenomena, e.g., wildlife [1] and phytoplankton [2] distribution. Spatial field modeling is important for many applications, such as environmental monitoring and precision agriculture [3].

Mainstream adaptive sampling strategies commonly exploit Gaussian Process Regression (GPR) [4]—a non-parametric Bayesian model, to estimate the distribution of an environmental phenomena from its samples—because of its effectiveness in modeling spatial fields and providing an uncertainty measure for identifying informative regions [5]–[9]. However, GPRs typically assume a homogeneous distribution of the spatial field of interest, an assumption that breaks when modeling non-stationary fields as there tends to be heterogeneous spatial distributions [10]. In addition, due to the cubic time computational complexity of GPR models, their application in environments that require a large number of measurements is still prohibitive. To address these limitations, some work has been done to enhance the performance of GPRs and its flexibility in modeling uneven variations. Examples include sparse GPRs [11], [12] that regulate the input data while maintaining high performance; multi-kernel learning [13] that combine several kernels to enhance the flexibility of the covariance function in modeling uneven distributions; methods that project the non-stationary field into a different space so that the stationary assumption can hold [14]; and a mixture of experts (MOE) [15]–[17] approach. Despite these efforts, the practical use of such enhanced modeling methods for adaptive sampling with a desired predictive accuracy is still an open challenge.

This paper proposes a scalable adaptive sampling strategy for multiple robots, which aims at modeling non-stationary spatial fields with a specific predictive accuracy, while minimizing traveled distance. To achieve these goals, taking inspiration from MOE, we propose a network of GPR experts, which are incrementally created by a team of robots as they sample the environment. Each expert is associated with a particular non-overlapping sub-region that it is tasked to model. Each robot creates its own sub-network of experts and uses it to determine where to collect samples and when a given expert can model its corresponding sub-region. Robots coordinate by restricting themselves to visiting a particular subarea of the environment through sharing the regions covered by the sub-network. A robot creates a new expert by visiting and sampling data at a location that is not covered by any expert. Depending on the variation in the data collected in a given locality, a sparse or dense network of experts will be created, as shown in Fig. 1. Regions with less variance in the data will be modeled by fewer experts than those with high variance. Consequently, some regions are sampled more densely than others, which makes the sampling process

*This work was partially supported by NSF CNS-1919647, 2144624, OIA-1923004.
Department of Computer Science, Dartmouth College, Hanover, NH USA {kizito.masaba.gr, alberto.quattrini.li}@dartmouth.edu

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. DOI: Accepted at IEEE MRS 2023
adaptive to the non-stationarity in the environment.

Overall, this paper makes the following contributions:
1) a novel distributed MOE-based approach to modeling non-stationary spatial fields with a desired predictive accuracy.
2) a novel adaptive sampling strategy that reasons on such a model, allowing robots to identify informative sampling locations to explore.
3) a ROS implementation and an experimental analysis with realistic simulations comparing other methods used for environmental monitoring. Results show that enforcing a desired predictive accuracy enhances modeling and overall adaptive sampling performance.

II. RELATED WORK

Different approaches have been proposed for the sampling problem, where robots have to collect spatial information in the environment. Some recent surveys on exploration and sampling include [18], [19]. These methods can be categorized into two main families: non-adaptive and adaptive.

In non-adaptive approaches, informative paths are computed offline using a previously trained model, if available, and shared among robots. The problem addressed by these approaches is also called coverage problem [20]–[22], where the robots are assumed to have a sensor footprint and they need to cover every point in the workspace. Such coverage algorithms typically decompose the environment into cells, abstract them through a graph, and solve the related problem of visiting every node – please see a comprehensive survey from Galceran et al. [22] for full details. Some offline methods have looked at how to determine informative trajectories for modeling spatial distributions, for example, using a sequential allocation approach to find paths to allocate to robots, with mutual information to measure their informativeness [23]; an asymptotically optimal RRT-based approach [24]; or trained models based on data collected beforehand to find offline informative paths with a certain minimum accuracy [25].

Non-adaptive approaches have the advantage that plans can be computed beforehand without being constrained by on-board computational power availability. In fact, they are the main approach used in environmental monitoring missions. The main limitation of these methods is that computed plans do not adapt according to the collected data during the mission, which may introduce inefficiencies. In contrast, our method explicitly adapts the plan according to the data collected by the robots at a given locality in the environment.

In adaptive approaches, robots learn the spatial field by continuously updating their belief of the field using the collected samples and re-planning informative paths. Various works [6], [26]–[33] used different metrics to measure uncertainty (e.g., variance, entropy) and proposed information-based acquisition functions that enable robots to select informative locations to sample, which are locations that maximize a given acquisition function.

These approaches are capable of adapting the plan to the collected data and drive robots to areas of higher uncertainty. They typically use a single model for the entire environment. In practice, the choice of using a single model over the whole environment has two limitations: 1) the scalability in the size of the environment might be limited and 2) there might be variations in the environment that would be better captured by an ensemble of models.

The scalability of the model has been addressed by some methods [7], [34], [35] that reason on smaller partitions of the environment. Some works [36], [37] proposed data fusion strategies that minimize the data used in training the GP. No guarantee on the predictive accuracy is provided by adaptive methods. Our work proposes a solution that focuses on maintaining a defined predictive accuracy based on the real-time data stream.

A specific type of model that can help towards improving the predictive accuracy is the Mixture of Gaussian Processes (MGP) that was introduced by Tresp et al. [15] as a model for active learning – another name for adaptive sampling – [38] problems. These methods model large spatial fields by aggregating local approximations obtained from a collection of models. For a comprehensive review of MGPs, please see the surveys from Liu et al. [39] and Yuksel et al. [17]. Here, we highlight some of the existing work that used MGPs in multi-robot adaptive sampling. For instance, Ouyang et al. [8] proposed a decentralized MGP approach for multirobot active learning of non-stationary phenomena, where each robot represents a Gaussian Process Expert (GPE) and data points are assigned to each expert following a Dirichlet Process [40]. Luo et al. [9] presented a distributed mixture of GPs that aim at improving the tractability of GPs in a multirobot system by eliminating the need for sharing raw data between robots. This is achieved by enabling each robot to model its own collected data and only exchange the resulting optimal GP parameters with other robots. Each robot creates one GP expert that is linked to other experts through their respective optimal parameters. Similarly, Kemna et al. [7] and Fung et al. [41] developed a decentralized mixture of GP models, where each robot has one model. To improve coverage and tractability of the GP models, robots independently divide the environment into Voronoi partitions [42], [43], which are then assigned to each robot. These approaches use information-theoretic acquisition functions, such as entropy, mutual information [6], [44], Integrated Variance Reduction (IVR) [28] and Upper Confidence Bound (UCB) [44] to identify hot-spots for sampling.

Our work differs from the existing work in 3 main aspects: 1) we define a new acquisition strategy whose utility is a function of a network of GP experts, which cover the explored region. This eliminates the data redundancy problem that results from the use of most acquisition functions such as IVR, UCB, and entropy [45], and greatly reduces the cost of computing sampling locations; 2) the number of experts required to model and accordingly partition a given workspace is independent of the size of the robot team – it only depends on the variation in the target environmental phenomenon; 3) we enforce a desired predictive accuracy on
the local estimates of every sub-region.

III. BACKGROUND: GAUSSIAN PROCESS REGRESSION

As our proposed approach is based on GPs and their properties, here we include a brief description of the GPs together with an analysis on the accuracy of the estimates. Let \( X \) be a 2D vector of locations and \( y \) a vector of their corresponding measurements collected by the robot. \( X \) is a vector of test locations, whose measurements are to be estimated. Then, a GP model \( f \) for estimating \( X \) is drawn from a normal distribution defined as

\[
f \sim N(\mu(X, X), \Sigma),
\]

where the mean vector \( \mu(X, X) \) and covariance matrix \( \Sigma \) are

\[
\mu(X, X) = K(X, X)[K(X, X) + \sigma_n^2 I]^{-1} y,
\]

\[
\Sigma = K(X, X) - K(X, X)[K(X, X) + \sigma_n^2 I]^{-1}K(X, X).
\]

The elements of the covariance matrix, \( K(\cdot, \cdot) \) are given by a kernel function, which describes the spatial correlation between a pair of locations. We use a commonly used kernel because of its general applicability to different domains, the squared exponential (SE) [4], defined as

\[
k(x_p, x_q) = \sigma_f^2 \exp(-\frac{(x_p - x_q)^2}{2l^2}) + \sigma_n^2 \delta_{pq},
\]

where \( l \) is the length scale representing the function smoothness; \( \sigma_f^2 \) is signal variance determining the amplitude; \( \sigma_n^2 \) is the noise variance accounting for the estimate noise; and \( \delta_{pq} \) is the Kronecker delta (\( \delta_{pq} = 1 \) if \( p = q \), else \( \delta_{pq} = 0 \)).

Using the SE kernel, a GP model is parameterized by \( \theta = (\theta_f^2, l, \theta_n^2) \), which are determined from the data using Maximum Likelihood Estimation (MLE) [4], by maximizing

\[
\log p(y|X, \theta) = -\frac{1}{2} y^T \Sigma^{-1} y - \frac{1}{2} \log |\Sigma_y| - \frac{n}{2} \log 2\pi.
\]

Suppose we are given a GPR estimator, \( \hat{\theta} \) that estimates some random variable \( \theta \); the Mean Squared Error (MSE) [40] in the estimates is given by

\[
\text{MSE}(\hat{\theta}) = E[(\hat{\theta} - \theta)^2]
\]

and we can express \( E[(\hat{\theta} - \theta)^2] \) in terms of the GP variance to obtain [40]

\[
\text{MSE}(\hat{\theta}) = \text{Var}(\theta) + \text{Bias}^2(\hat{\theta})
\]

where \( \text{Var}(\hat{\theta}) \) is the posterior variance of GPR and \( \text{Bias}(\hat{\theta}) \) is the model bias. Accordingly, for an unbiased estimator as a GP [25]

\[
\text{Var}(\hat{\theta}) \leq \text{MSE}(\hat{\theta}) = \Delta
\]

From Equation [40], we can define the predictive accuracy of GPR estimates as a function of its posterior variance. Based on this formulation, we define a threshold, \( \Delta \), that ensures the corresponding desired predictive accuracy.

Note that the running time and memory complexity of GP modeling is \( O(N^3) \) and \( O(N^2) \), respectively [47] – where \( N \) is the size of the training set \( (X, y) \). This makes the GP model intractable for real-time exploration of large areas (e.g., \( N > 100 \)) on an embedded system as a Raspberry Pi.

IV. PROBLEM STATEMENT

Given a 2D environment, \( \mathcal{E} \) with an unknown non-stationary spatial field, \( \mathcal{S} \), such that \( s = \{x \in \mathcal{E}, z \in \mathcal{R}\} \in \mathcal{S}, \forall x \in \mathcal{E} \); a team of \( K \) differential drive robots, \( R = \{1, ..., K\} \) that can communicate with each other through long-range WiFi or radio devices, and are equipped with a sensor that measures \( z \). The robots’ task is to adaptively sample from the environment to reconstruct \( \mathcal{S} \) with a minimum desired predictive accuracy \( \Delta \) and minimum traveled distance.

Although \( \mathcal{S} \) is unknown, we can assume that it is composed of an unknown collection of homogeneous sub-fields so that spatially heterogeneous fields can be modeled. Hence, \( \mathcal{E} \) can be expressed as \( \mathcal{E} = \bigcup_{i=1}^{\infty} v_i \), where \( v_i \subseteq \mathcal{E} \) is a sub-region centered at \( x_i \) within a radius \( r_i \), containing a homogeneous subfield, \( \mathcal{S}_i \subseteq \mathcal{S} \). For simplicity, we truncate \( v_i \) such that it is bounded by a square that is circumscribed by the disc of radius \( r_i \). This simplification allows us to work with grids, a popular discretization approach for coverage and exploration problems. Therefore, \( v_i \) is bounded by a square shaped region of length, \( L_i = r_i\sqrt{2} \). Future work will look at other discretizations.

With sufficient measurements, \( M_i \subseteq \mathcal{S}_i \) collected from within \( v_i \), \( \mathcal{S}_i \) can be reconstructed by a regression model, \( m_i \) with a desired predictive accuracy, \( \Delta \). However given that the boundaries of \( S_i \) are unknown and are potentially non-linear, it is also necessary to include measurements collected from the neighborhood of \( v_i \). This allows for accurate modeling of boundary regions of \( S_i \).

We define a neighbor, \( v_j \) of \( v_i \) as any sub-region such that some of its measurements are within distance \( \frac{r_j}{2} \) with at least one location, \( x_i \in v_i \) (i.e., \( \exists x_i \in v_i, x_j \in v_j \); \( ||x_i - x_j|| \leq \frac{r_j}{2} \)). To model \( S_i \), all locations in \( v_i \) must have a neighbor sub-region. Henceforth, we refer to this condition as the neighborhood constraint. For example, if all sub-regions are approximately of equal area (as in grids), then \( v_i \) will need to have measurements collected around the center of its surrounding sub-regions (between 3 and 8 neighbors of approximately equal sized grid cells) in order to satisfy the neighborhood constraint.

Identifying all the sub-regions that make up \( \mathcal{S} \) along with their neighbors such that each sub-region satisfies its neighborhood constraint allows the robots to achieve the adaptive sampling task.

V. RAPIDLY-SAMPLING ADAPTIVE GRAPH

We devise the Rapidly-sampling Adaptive Graph (RAG), a connected undirected graph \( \mathcal{G} = (V, E) \) (with \( V \) nodes and \( E \) edges), that encodes the predictive ranges of models used for estimating various subfields and the boundary neighborhood of these subfields. A subfield is a contiguous, homogeneous subset of a spatial field. A node \( v_i = (x_i, L_i, m_i) \in \mathcal{V} \) is a subfield defined by a tuple consisting of a center location, \( x_i \in \mathcal{E} \), length, \( L_i \in \mathcal{R} \) of a boundary within which the subfield is enclosed and a regression model, \( m_i \), that can accurately estimate the same subfield with a desired predictive accuracy, \( \Delta \). In addition, \( (v_i, v_j) \in \mathcal{E} \) is an edge
Step 3: From the outputs we use $\sigma^2_{A|x}$ of each location to compute $L$ as shown in the following:

$$L = \sqrt{2} \max\{\min(x - x_D)\forall x \in A, \forall x_D \in V \} \text{ s.t. } \sigma^2_{A|x} \leq \Delta$$

which is the maximum distance of every location in $A$ to its closest measurement. The rationale of this formulation is based on the property of the SE kernel, which we apply in GPR modeling. With a SE kernel, posterior variance/correlation of any location, $x$, to other estimates and measurements is only dependent on the hyper-parameters obtained from $D$ and the distance between $x$ and these locations, which reduces exponentially as the distance increases. Therefore, the estimate at a given location is highly dependent on its closest measurement. We hereby attribute this relationship to the predictive power of GPR, which is how wide it can accurately estimate a measurement at an unobserved location in $A$. Hence, we can observe that a GPR model that is trained with data in some region $A$, centered at $x$ can predict a certain subset of the locations within $A$ with a desired predictive accuracy, $\Delta$.

Although the area of $A$ can be arbitrary, its size has to be carefully selected to balance the trade-off between performance and predictive power of the model, which inherently balances the exploration-exploitation trade-off. In this work, we consider $A$ to be a square region of length $2L$.

2) RAG Edge Generation: When a new RAG node is created, it induces new edges on the RAG depending on its proximity to other nodes and the boundary length of each node.

A. RAG Construction

RAG construction is an iterative two-step process composed of RAG node generation and RAG edge generation, until the full RAG is constructed – see Fig. 2.

Algorithm 1 RAG Sampling Algorithm

**Input:** $x_0, L_0$: Initial Center and Boundary Length

1. $R, x, L \leftarrow 0, x_0, L_0$
2. while $x \neq -1$ do
3. Navigate to $x$
4. $A \leftarrow$ Extract Sub-region of square length $2L$ centered at $x$
5. $D \leftarrow$ Get measurements in square of length $L$ centered at $x$
6. $v \leftarrow$ CreateRAGNode($x, D, A$)
7. AddNodeToRAG($R, v$)
8. $v$.status $\leftarrow$ COVERED
9. ModelCompleteNodes($R$)
10. ShareRAGNode($v$)
11. $x \leftarrow$ FindNextCenter($R, x$)
12. $L \leftarrow$ GetNextBoundaryLength($R, x$)
13. end while

**Algorithm 2**

1) RAG Node generation: Suppose we have a location $x$ in the environment. To obtain the length of the boundary $L$, a GPR model is used to estimate the measurements of every location in some arbitrary area, $A$ that is centered at $x$ using data, $D$ collected from within $A$. The output of a GPR prediction is the measurement estimates (posterior mean, $\mu_{A|x}$) and the predictive uncertainty (posterior variance, $\sigma^2_{A|x}$) of each location in $A$ – as discussed in Section [III](#SectionIII).

With the environment initially unexplored, we propose the following strategy for a robot in pose $x_r$, to find the next location to sample:

$$x_{opt} = \arg \min \{||x_r - x|| \mid I(x), \forall x \notin V \}$$

The overall objective is to minimize traveled distance and at the same time to ensure that created RAG nodes quickly satisfy their neighborhood constraints and get modeled while extending to unexplored regions. Thus, we define the informativeness $I(x)$ of $x$ in an unexplored (i.e., it is not covered by the RAG) sub-region of the workspace, as $I(x) = \{v_i : ||x - x_i|| \leq L_i \land x \notin v_i\}, \forall i = 1, ..., |V|$, which is the...
number of RAG nodes that are close enough to $x$ that they can form a RAG edge with a node created at $x$. Specifically, we need to find a location in the unexplored sub-region at which a new node can be created and linked to at least one RAG node.

C. Modeling RAG Nodes

A sub-region of a given RAG node is modeled when the neighborhood constraint is satisfied by the node. A RAG node that satisfies such a constraint is called a complete node. A complete RAG Node, $v_i$ has at least one neighbor on the RAG, and each neighbor has its own model and data.

To model $v_i$, we need to compare the data obtained from its neighbors against its own data in order to ascertain if they are all homogeneous (i.e., have similar distributions) and thus determine whether its model needs retraining. If data from the neighborhood share a similar distribution with that of $v_i$, then $v_i$ model needs no retraining. Otherwise, it will be retrained with the neighboring data in order to learn the patterns in the neighborhood, before the neighboring data is incorporated into the modeling. We perform a pairwise Student’s t-test [48] to determine the similarity between $v_i$’s data and that from each of its neighbors.

The new modeling dataset $D^*_i$ for the $v_i$ model is the data from $v_i$ and all its neighbors:

$$D^*_i = D_i \cup \bigcup_{j=1}^{p} D_j$$  \hspace{1cm} (11)

Using $D^*_i$, the $v_i$ model is used to estimate the distribution of every location within the $v_i$ node, which is the ultimate spatial field estimate of this sub-region. The status of the modeled node is then changed to MODELED.

D. Robot Coordination

The RAG allows multirobot task allocation, restricting robots to particular regions of the environment. Initially, robots can be deployed in arbitrary locations. However, as they seek to construct their respective RAG, they encounter boundaries along the way.

We define boundaries as regions, which have been explored, yet they are not covered by the RAG. These regions are flagged by other robots through communication. A robot shares new nodes of its RAG with communicating robots to increase the situation awareness of its surrounding robots. Note that the algorithm does not require full communication since robots can continue expanding the RAG even if they are unable to communicate.

Note that RAG construction is restricted to unexplored regions and the planning objective function prioritizes locations that are closer to a robot’s RAG nodes, thus pushing robots within their boundaries.

VII. Experimental Results and Discussion

Setup. We implemented our method in Python and the Robot Operating System (ROS) [49] to enable portability across different robotic platforms. We ran several experiments with teams of differential drive autonomous surface vehicles (ASVs) in 2D and 3D simulators – Stage [50] and Gazebo [51] respectively – that have actuators and localization noise. In addition, Gazebo includes water dynamics.

We simulated four different environments, each with a generated unique non-stationary spatial field (SF) (Fig. 3), representing realistic variations of Chlorophyll-a concentration as an indicator of the presence of Cyanobacteria [52] in lakes. The purpose for these variations is to evaluate the adaptiveness of the proposed method in environments with various phenomena. Environments with SF1, SF2 and SF3 cover an area of size $400 \times 400$ m whereas the environment with SF4 cover an area of size $823 \times 134$ m. We implemented a simulated sensor providing SRF measurements with Gaussian noise to account for the sensor error, such as the YSI EXO 2 sonde with chlorophyll-a probe [53], so that we can analyze the robustness to noisy data.

We spawned ASV teams of 3 different sizes 2, 4, 6. We ran experiments with three (3) different accuracy thresholds, $\Delta$ to assess its effect on the overall performance, namely; 0.5, 1.0 and 1.5. Each experiment was repeated 5 times. The experiments are run on a Ubuntu 20.04 computer with an Intel i7 CPU with 32GB RAM.

For comparison with our method (labeled RAG with corresponding $\Delta$), we included both non-adaptive – being a commonly used strategy – and adaptive alternatives. For non-adaptive, we implemented a lawn-mower way point profile (labeled as LM), with five (5) different inter-lap spacing, namely; 8 m, 10 m, 20 m, 30 m and 40 m. We set 8 m resolution because it has been shown to perform better than higher resolutions [25] when a boustrophedon [54] waypoint pattern is applied. For adaptive, we implemented the dynamic partitioning approach (labeled as DPart with corresponding $\Delta$) [7], where a robot team iteratively divides the environment into Voronoi partitions and each robot samples one unique partition. Instead of using only time as the termination criterion, we modify DPart to terminate whenever a desired $\Delta$ is achieved. We also cap the mission time to 3500 s as applied in [7] to guarantee termination.

**Qualitative Analysis.** As sample representatives, Fig. 4 shows the reconstructed fields of each environment, for RAG0.5, LM_08, LM_40 and DPart0.5 with a team of 6 robots respectively. We observe that all methods except DPart0.5 accurately modeled SF1 and SF2. However, only RAG0.5.
managed to reconstruct $SF3$ accurately. The failure for both $LM_*$ and $DPart0.5$ is due to the areas that are not sampled by the robots. $LM$ paths are planned with a non-adaptive interlap spacing, which limits the informativeness of the path and might miss for example the higher spot in $SF3$. On the other hand, $DPart$ dynamically partitions the environment such that each robot focuses on a particular partition. As a result, some subregions of the environment can be ignored. These two shortfalls do not affect $RAG$ as it ensures that all regions of the environment are sufficiently sampled in order to achieve the desired predictive accuracy.

Indeed, looking at the robot trajectories in the four environments – Fig. 5 – we observe that robots sample the fields more densely along the boundary of the two heterogeneous regions compared to the homogeneous regions. This is due to the high variance in the data collected in this area.

**Quantitative Analysis.** The quantitative evaluation is in terms of predictive accuracy measured as average posterior variance (aPVAR), reconstruction error measured as Root Mean Squared Error (RMSE) [53], average distance traveled by the robots, and computation time to assess the feasibility of a given method for use in online adaptive sampling. As a representative example (other environments overall follow similar trends) we show the results for $SF2$ in Fig. 6.

The GPR estimates in $RAG$ variants have the least aPVAR across all robot teams and environments because of the $\Delta$. This is followed by $LM$ and then $DPart$ variants. Both $RAG$ and $LM$ variants record lower reconstruction error across all robot teams than $DPart$ especially with a team of 2 robots. Observing its trajectories, $DPart$’s approach of partitioning the environment tends to leave out some unexplored regions. Overall, $RAG$ variants achieve the least traveled distance across all robot teams, except for $LM_{30}$ and $LM_{40}$, which however have the worst learning accuracy. The low traveled distance of our method is attributed to the adaptiveness of the data collection process as shown in the qualitative analysis. The computation time results show a constant time for $RAG$.

**VIII. Conclusion**

This work proposes an efficient multirobot active learning strategy for accurately reconstructing spatial fields. We propose an approach that allows for continuous modeling of the desired phenomenon with locally collected data to achieve local optimality based on a predefined desired predictive accuracy, $\Delta$. The locally optimal regions are interconnected to account for local discontinuities, such that a given local region is modeled only when its entire neighborhood has been sampled. We model this process as a rapidly sampling adaptive graph, where its nodes are the locally optimal regions and edges encode the neighborhood of each locally optimal region. Experiments and results show a superior performance of the proposed method over non-adaptive and adaptive strategies from the literature.

In future work, we plan to incorporate kinodynamic constraints to account for other robot types for generating efficiently reachable neighbors, as well as explicitly modeling communication constraints to enable robots to take decisions on when to share information with other robots. In addition, we plan to study long-term sampling where temporal resolution is another important factor to decide when to deploy the robots. Finally, we intend to test the
proposed strategy in the real world, e.g., lake, to study algal blooms with a team of ASVs and contribute towards high-impact applications, such as the study of climate change.

REFERENCES


