Abstract—In this article, we study the problem where a group of agents aims to collaboratively learn a common static latent function through streaming data. We propose a lightweight distributed Gaussian process regression (GPR) algorithm that is cognizant of agents’ limited capabilities in communication, computation, and memory. Each agent independently runs agent-based GPR using local streaming data to predict test points of interest; then, the agents collaboratively execute distributed GPR to obtain global predictions over a common sparse set of test points; finally, each agent fuses results from distributed GPR with agent-based GPR to refine its predictions. By quantifying the transient and steady-state performances in predictive variance and error, we show that limited interagent communication improves learning performances in the sense of Pareto. Monte Carlo simulation is conducted to evaluate the developed algorithm.

Index Terms—Distributed algorithms, Gaussian processes, machine learning.

I. INTRODUCTION

etworks of agents can access large amount of streaming data online in many applications, e.g., autonomous driving [1] and precision agriculture [2]. Machine learning has been increasingly adopted to extract reliable and actionable information from Big Data, and enable agents to adapt and react in uncertain and dynamically changing environments. Nowadays, machine learning algorithms have achieved remarkable performances in terms of consistency [3], generalization [4], and robustness [5]. However, limited resources challenge implementation of the algorithms on physical agents, e.g., mobile robots and Internet-of-Things devices.

Gaussian process regression (GPR) [6] is an efficient nonparametric statistical learning model. GPR models a target function as a sample from a Gaussian process prior specified by a pair of mean and covariance functions only depending on inputs. With proper choice of prior covariance function, also referred as kernel, and mild assumptions on the target function, GPR is able to consistently approximate any continuous function [7]. With optimal sampling in the input space and covariance functions obeying Sacks–Ylvisaker conditions of order $r$, the generalization error of GPR diminishes at the rate of $O(n_s^{-(2r+1)/(2r+2)})$ (see Section V-B, [8]), where $n_s$ is the number of samples. GPR is able to quantify learning uncertainties since it predicts the target function in the form of a posterior probability distribution. GPR has demonstrated powerful capabilities in various applications, e.g., optimization [9], motion planning [10], and trajectory estimation [11].

GPR scales as $O(n_s^3)$ in computational complexity and $O(n_s^2)$ in memory [6, p. 171], which prohibits applications with large datasets. There are multiple sparse approximation methods for large datasets. One major class of approximation methods, which is also referred as global approximation, tackles the computational complexity by achieving the sparsity of the Gram matrix. Methods include using a subset of data to approximate the whole training dataset, designing a sparse kernel, and sparsifying the Gram matrix. The best possible result can be achieved by global approximation algorithms is $O(n_s^2)$ in computational complexity and $O(n_s^2)$ in memory, where $m_s \ll n_s$ is the number of inducing points or the size of a subset of training data. More details about global approximation can be found in the recent survey paper [12]. In the community of geostatistics, nearest-neighbor GPR [13] is applied [14], [15], where the predictions are made only using the training data of the nearest input. It requires only $O(n_s)$ in both memory and (worst-case) computation.

Centralized implementation of GPR is not suitable for networks of agents due to poor scalability in data size, high cost in communication and memory, and fragility to single-point failures. There have been studies on distributed GPR over server–client architecture, which is also referred to as divide-and-conquer approach or local approximations [12]. In the server–client architecture, a server acts as the centralized entity that partitions a dataset and assigns each subset of the data to computing units (clients). The clients perform training independently and send their learning results to the server for postprocessing. These methods speed up the training process and are able to scale to arbitrarily large datasets. Communication budget constraint is considered in [16] by reducing the dimensionality of transmitted data to approximate the whole dataset. Sparse approximation of full GPR is used in [16] to further relieve the communication overhead. Notice that the server–client architecture requires each client being well connected with the server, and is not robust to the failure of the server. Chen et al. [17] decentralized sparse approximations of full GPR for fixed datasets over complete communication graphs. A distributed algorithm is also proposed to deal with fixed and sparse graphs. However, this article considers offline learning with static datasets on the agents and does not provide theoretic guarantee on the distributed algorithm.
Our work is related to multi-agent regression using kernel methods and basis functions. The authors in [18], [19], and [20] studied offline learning, where all training data are provided before learning, using kernel methods. The authors in [21], [22], and [23] studied online learning, where training data are collected successively by mobile robots, using basis functions. In particular, they approximate the unknown functions with a linear combination of a finite number of known basis functions. This reduces the problem into a parameter estimation problem. From the perspective of regression, the problem investigated in [21] is equivalent to selecting the centers of a finite number of basis functions defined by Voronoi partition. In contrast, this article considers online learning of abstract agents using Gaussian processes, where the unknown function is modeled as a sample from a distribution of functions.

**Contribution statement:** We consider the problem where a group of agents aims to collaboratively learn a common static latent function through streaming data. We propose lightweight distributed Gaussian process regression (LiDGPR) algorithm for the agents to solve the problem. More specifically, each agent independently runs agent-based GPR using local streaming data to predict the test points of interest; then, the agents collaboratively execute distributed GPR to obtain global predictions over a common sparse set of test points; finally, each agent fuses the results from distributed GPR with those from agent-based GPR to refine its predictions. Our analysis of the transient and steady-state performances in predictive variance and error reveals that through communication agents whose data samples have lower dispersion (or observation noise has lower variance) help improve the performance of the agents whose data samples have higher dispersion (or observation noise has higher variance). The improvements in learning performances are in the sense of Pareto, i.e., some agents’ performances improve without sacrificing other agents’ performances. In summary, our major contributions are twofold:

1. We develop LiDGPR that is cognizant of agents’ limited capabilities in communication, computation, and memory.
2. We analyze the predictive mean and variance of LiDGPR and quantify the improvements of the agents’ learning performances resulted from interagent communication.

Monte Carlo simulation is conducted to evaluate the developed algorithm. In addition to the preliminary version of this paper [24], this version includes a new set of theoretical results on the steady-state performance of predictive errors, identifies the factors that affect the improvement of learning performances, provides the proofs of the theoretical results, and discusses how the algorithm is cognizant of limited resources. Due to space limitation, the proofs of some intermediate results are included in the complete version [25].

**Notations:** We use lower-case letters, e.g., $a$, to denote scalars, bold letters, e.g., $\mathbf{a}$, to denote vectors; we use upper-case letters, e.g., $A$, to denote matrices, calligraphic letters, e.g., $\mathcal{A}$, to denote sets, and bold calligraphic letters, e.g., $\mathbf{A}$, to denote spaces. For any vector $a$, we use $a_i$ to denote the $i$th entry of $a$. For any matrix $A$, we denote $a_{ij}$ as the entry at $i$th row $j$th column. Denote $I_n \in \mathbb{R}^{n \times n}$ the $n$-by-$n$-dimensional identity matrix, $1_n \in \mathbb{R}^n$ the $n$-dimensional column vector with all 1’s, i.e., $[1, \ldots, 1]^T$, and $0_n$ analogously.

We use superscript $(\cdot)^{[i]}$ to distinguish the local values of agent $i$, and $(\cdot)^{\max}$ $(\cdot)^{\min}$ denote the maximum (minimum) of the local values, e.g., $a^{\max} \triangleq \max_{i \in \mathcal{V}} a^{[i]}$. We denote superscript $(\cdot)^T$ the transpose of a vector or matrix, and bracket $[\cdot]_{\mathcal{E}}$ the column vector with elements satisfying event $\mathcal{E}$. Denote $\mathbb{E}_a[\cdot]$ the expectation taken over the distribution of random variable $a$, and $P(\cdot)$ a distribution. We use $O(\cdot)$ to denote the conventional big $O$ notation, i.e., $O(g(t))$ represents the limiting behavior of some function $f(t)$ if $\lim_{t \to \infty} \frac{f(t)}{g(t)} = a$ for some constant $a > 0$.

We use $\geq (\leq)$ to denote elementwise comparison between two vectors, i.e., for any $a,b \in \mathbb{R}^n$, $a \geq (\leq) b$ if and only if $a_i \geq (\leq) b_i$ for all $i = 1, \ldots, n$. Operation $|a|$ takes the absolute values elementwise on vector $a$, $|A|$ returns the cardinality of set $A$, and $|a|_{\infty} \triangleq \max_j |a_j|$ for any vector $a$. Define the distance metric $\rho(z, z') \triangleq \|z - z'\|$, the point to set distance as $\rho(z, \mathcal{Z}) \triangleq \inf_{z' \in \mathcal{Z}} \rho(z, z')$. Define proj$(z, \mathcal{Z}) \triangleq \{z' \in \mathcal{Z} | \rho(z, z') = \rho(z, \mathcal{Z})\}$ the projection set of point $z$ onto set $\mathcal{Z}$. Denote the supremum of a function $\eta$ as $\|\eta\|_{\mathcal{Z}} \triangleq \sup_{z \in \mathcal{Z}} |\eta(z)|$.

## II. PROBLEM STATEMENT

**Network model:** Consider a network of agents represented by a directed time-varying communication graph $G(t) \triangleq (\mathcal{V}, \mathcal{E}(t))$, where $\mathcal{V} \triangleq \{1, \ldots, n\}$ represents the agent set, and $\mathcal{E}(t) \subseteq \mathcal{V} \times \mathcal{V}$ denotes the edge set at time $t$. Notice that $(i, j) \in \mathcal{E}(t)$ if and only if agent $i$ can receive messages from agent $j$ at time $t$. Denote the set of the neighbors of agent $i$ at time $t$ as $\mathcal{N}^t(i) \triangleq \{j \in \mathcal{V} : (i, j) \in \mathcal{E}(t), \text{ and } j \neq i\}$. The matrix $A(t) \in \mathbb{R}^{n \times n}$ represents the adjacency matrix of $G(t)$ where $a_{ij}(t) \neq 0$ if $(i, j) \in \mathcal{E}(t)$. We make the following standard assumptions [26] about the network topology.

**Assumption II.1. (Periodical strong connectivity):** There exists positive integer $b \geq 1$ such that, for all time instant $t \geq 0$, the directed graph $(\mathcal{V}, \mathcal{E}(t) \cup \mathcal{E}(t+1) \cup \cdots \cup \mathcal{E}(t+(b-1)))$ is strongly connected. This guarantees the information of each agent can reach any other agents in the network within finite time.

**Assumption II.2. (Balanced communication):** It holds that $1_n^T A(t) = 1_n^T$ and $A(t)1_n = 1_n$, for all $t \geq 0$.

In the consensus literature, the first part of Assumption II.2 is called column stochasticity and is a standard sufficient condition to reach consensus. The second part is called row stochasticity and is needed to guarantee average consensus.

**Assumption II.3. (Nondegeneracy):** There exists a constant $\alpha > 0$ such that $a_{ii}(t) \geq \alpha$ and $a_{ij}(t) \in \{0\} \cup [\alpha, 1]$, for all $t \geq 0$.

That is, each agent assigns nontrivial weights on information from itself and its neighbors.

**Observation model:** At each time instant $t$, each agent independently observes the outputs of a continuous common static latent function $\eta : \mathcal{Z} \rightarrow \mathcal{Y}$ with zero-mean Gaussian noise, where $\mathcal{Z} \subseteq \mathbb{R}^n$ is the compact input space for $\eta$. The observation model is given by

$$y^{[i]}(t) = \eta(z^{[i]}(t)) + \epsilon^{[i]}(t), \quad \epsilon^{[i]}(t) \sim \mathcal{N}(0, \sigma^{[i]}_e^2)$$

(1)

where $z^{[i]}(t) \in \mathcal{Z}$ is the input of $\eta$ from agent $i$ at time $t$, $y^{[i]}(t) \in \mathcal{Y}$ is the observation of agent $i$, and $\epsilon^{[i]}(t)$ is independent Gaussian noise. Note that we do not assume that input $z^{[i]}(t)$ follows any distribution, which is a standard assumption in statistical learning [3]. We let $\eta(\mathcal{Z})$ return a column vector $[\eta(z)]_{z \in \mathcal{Z}}$, and similarly for other functions. For notational simplicity, it is assumed that the output space $\mathcal{Y} \subseteq \mathbb{R}$ because multidimensional observations can always be decomposed as aggregation of 1-D observations.
Problem Statement: The objective of this article is to design a distributed algorithm for the agents to learn the common static latent function \( \eta \) via streaming data \( \{y[i](t), z[i](t)\}_{t=1}^{\infty} \). The challenges of the problem stem from the fact that the training dataset is monotonically expanding due to incremental sampling while the agents have limited resources in communication, computation, and memory.

The followings are examples of potential applications of this formulation. One example can be a group of mobile robots deployed in a vast open area to collaboratively monitor a static signal, such as temperature or wind field (see the case study in Section VI). Other examples include the learning of the dynamics of a moving target using a network of static sensors [27]. In addition to robotic applications, this formulation also applies to profit predictions in marketing and wheat crop prediction [28].

III. PRELIMINARIES

In this section, we provide necessary background on GPR. Let \( \eta : \mathcal{Z} \rightarrow \mathbb{R} \) be the target function, where \( \mathcal{Z} \subseteq \mathbb{R}^{n_z} \). Given input \( z(t) \in \mathcal{Z} \) at time \( t \), the corresponding output is: \( y(t) = \eta(z(t)) + \epsilon(t), \epsilon(t) \sim \mathcal{N}(0, \sigma_e^2) \), where \( \epsilon(t) \) is the Gaussian measurement noise. Let training data be in the form \( D \triangleq (z, y) \), where \( \mathcal{Z} \triangleq \{z(1), \ldots, z(n_z)\} \) is the set of input data and \( y \triangleq [y(1), \ldots, y(n_z)]^T \) is the column vector aggregating the outputs. GPR aims to estimate the function over a set of test data points \( \mathcal{Z}_t \subset \mathcal{Z} \) using \( D \) by modeling \( \eta \) as a sample from a Gaussian process prior.

Definition III.1 ([6, p. 13]): A Gaussian process is a collection of random variables, any finite number of which has a joint Gaussian distribution.

Define kernel function \( k : \mathcal{R}^{n_z} \times \mathcal{R}^{n_z} \rightarrow \mathcal{R} \) that is symmetric and positive semidefinite; i.e., \( \int k(z, z')f(z')dv(z)dv(z') \geq 0 \) for all \( f \in L^2(\mathcal{Z}, \nu) \), where \( \nu \) denotes a measure ([6, p. 80]). By modeling \( \eta(\cdot) \) as a sample from the Gaussian process prior specified by mean function \( \mu(\cdot) \) and kernel \( k(\cdot, \cdot) \), the training outputs \( y \) and the test outputs \( \eta(\mathcal{Z}_t) \) are jointly distributed as

\[
\begin{bmatrix}
  y \\
  \eta(\mathcal{Z}_t)
\end{bmatrix} \sim \mathcal{N}
\begin{bmatrix}
  \begin{bmatrix}
    \mu(\mathcal{Z}) \\
    \mu(\mathcal{Z}_t)
  \end{bmatrix} \\
  \begin{bmatrix}
    K(\mathcal{Z}, \mathcal{Z}) + \sigma_e^2 I_{n_z} \\
    K(\mathcal{Z}_t, \mathcal{Z}) \quad K(\mathcal{Z}_t, \mathcal{Z}_t)
  \end{bmatrix}
\end{bmatrix},
\]

where \( K(\mathcal{Z}, \mathcal{Z}) \) returns a matrix such that the entry at the \( i \)th row and \( j \)th column is \( k(z(i), z(j)) \), \( z(i) \in \mathcal{Z}_t \), and analogously for \( K(\mathcal{Z}_t, \mathcal{Z}) \) and \( K(\mathcal{Z}_t, \mathcal{Z}_t) \).

Utilizing identities of joint Gaussian distribution ([6, p. 200]), GPR makes predictions of \( \eta \) on \( \mathcal{Z}_t \) based on dataset \( D \) as

\[
\eta(\mathcal{Z}_t) \sim \mathcal{N} \left( \begin{bmatrix}
  \mu(\mathcal{Z}) \\
  \mu(\mathcal{Z}_t)
\end{bmatrix},
\begin{bmatrix}
  K(\mathcal{Z}, \mathcal{Z}) + \sigma_e^2 I_{n_z} \\
  K(\mathcal{Z}_t, \mathcal{Z}) \quad K(\mathcal{Z}_t, \mathcal{Z}_t)
\end{bmatrix}^{-1}
\right)
\]

Algorithm 1: LiDGPR.

1: procedure
2: Input: network of agents: \( \mathcal{V} \); test inputs: \( \mathcal{Z}_t \); common inputs: \( \mathcal{Z}_agg \); adjacency matrix: \( A(t) \); prior mean function: \( \mu \); kernel function: \( k \); noise variance: \( (\sigma_e^2)^2 \) for \( i \in \mathcal{V} \).
3: \textbf{Init:} \( D(0) = \emptyset \), \( \xi[i](0) = 0 \), \( \mathcal{Z}_agg \).
4: for \( i \in \mathcal{V} \) do
5: \textbf{Agent-based GPR } \{\text{Algorithm 2}\}
6: \textbf{Distributed GPR } \{\text{Algorithm 3}\}
7: \textbf{Fused GPR } \{\text{Algorithm 4}\}
8: for \( t = 1, 2, \ldots \) do
9: do
10: \textbf{end for}
11: \textbf{end for}
12: \textbf{end procedure}

IV. LIGHTWEIGHT DISTRIBUTED GPR

In this section, we propose the LiDGPR algorithm, which allows the agents to collaboratively learn the static latent function subject to limited resources. As shown in Fig. 1, LiDGPR is composed of the following three parts.

1) Agent-based GPR (Algorithm 2), where the agents make their own predictions of \( \eta \) over a given set of points of interest \( \mathcal{Z}_t \subseteq \mathcal{Z} \) using local streaming data \( D[i](t) \triangleq (z[i](t), y[i](t)) \), where \( \mathcal{Z}[i](t) \triangleq \{z[i](1), \ldots, z[i](t)\} \) aggregates local input data and \( y[i](t) \triangleq [y[i](1), \ldots, y[i](t)]^T \) aggregates the outputs.

2) Distributed GPR (Algorithm 3), where the agents integrate their predictions with those of their neighbors on a predefined common set \( \mathcal{Z}_agg \subset \mathcal{Z} \), and estimate the predictions on this set given the global training dataset \( D(t) \triangleq \bigcup_{i \in \mathcal{V}} D[i](t) \).

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Algorithm 2: Agent-Based GPR.

1: procedure aGPR($\mathcal{D}^{[i]}(t)$)
2: for $z_s \in \mathcal{Z}$ do
3: choose $z_s^{[i]}(t) \in \text{proj}(z_s, \mathcal{Z}^{[i]}(t))$
4: $\tilde{\mu}_{z_s^{[i]}(t)} = \mu(z_s) + k(z_s, z_s^{[i]}(t))k(z_s^{[i]}(t), z_s^{[i]}(t))^{-1}(y^{[i]}_s - \mu(z_s^{[i]}(t)))$
5: $\tilde{\sigma}_{z_s^{[i]}(t)}^2 = k(z_s, z_s) - k(z_s, z_s^{[i]}(t))k(z_s^{[i]}(t), z_s^{[i]}(t))^{-1}k(z_s^{[i]}(t), z_s)$
6: end for
7: $\text{Return } \tilde{\mu}_{z_s^{[i]}(t)}, \tilde{\sigma}_{z_s^{[i]}(t)}^2$
8: end procedure

3) Fused GPR (Algorithm 4), where the agents refine the predictions on $\mathcal{Z}$ by fusing the results from distributed GPR with those from agent-based GPR.

The formal statement of LiDGPR is presented in Algorithm 1. For each iteration $t$, each agent $i$ collects data online and updates local dataset $\mathcal{D}^{[i]}(t) = \mathcal{D}^{[i]}(t-1) \cup (z_s^{[i]}(t), y^{[i]}_s(t))$, then, sequentially executes agent-based GPR, distributed GPR, and fused GPR.

A. Agent-Based GPR

To reduce computational complexity, we implement nearest-neighbor GPR as agent-based GPR. Instead of feeding the whole training dataset to full GPR in (2), agent-based GPR only feeds the nearest input(s) $z_s^{[i]}(t) \in \text{proj}(z_s, \mathcal{Z}^{[i]}(t))$, and the corresponding output $y^{[i]}_s(t)$, i.e., $(z_s^{[i]}(t), y^{[i]}_s(t))$, to (2) for each $z_s \in \mathcal{Z}$. If there are repeated observations over $z_s^{[i]}(t), y^{[i]}_s(t)$, the average of the observations. The predictive mean and variance for each $z_s$ are given in Lines 4 and 5 of agent-based GPR. Agent-based GPR returns $\tilde{\mu}_{z_s^{[i]}(t)}, \tilde{\sigma}_{z_s^{[i]}(t)}^2$ and $\tilde{\mu}_{z_s^{[i]}(t)}, \tilde{\sigma}_{z_s^{[i]}(t)}^2$, $\tilde{\mu}_{z_s^{[i]}(t)}$ and $\tilde{\sigma}_{z_s^{[i]}(t)}^2$, $\tilde{\mu}_{z_s^{[i]}(t)}, \tilde{\sigma}_{z_s^{[i]}(t)}^2$

B. Distributed GPR

Note that each agent only maintains $\mathcal{D}^{[i]}(t)$, a portion of the global training dataset $\mathcal{D}(t)$. Besides collecting more data, information exchanges between the agents could enhance the learning performance upon agent-based GPR. However, limited communication budget prevents the agents from sharing $\mathcal{D}^{[i]}(t)$, whose size monotonically increases. Hence, we develop distributed GPR where the agents communicate with the predictive means and variances over a common set $\mathcal{Z}_{agg}$.

In order to deal with large datasets using GPR, local approximation methods, such as product of expert (PoE) [29] and Bayesian committee machine [30], are proposed to factorize the training process. We consider the following PoE aggregation model for predicting each $z_s \in \mathcal{Z}_{agg}$:

$$\mu_{z_s^{[i]}(t)} = \frac{(\sigma_{z_s^{[i]}(t)}^2)}{\sum_{i=1}^{n} \sigma_{z_s^{[i]}(t)}^{-2} \tilde{\mu}_{z_s^{[i]}(t)}}$$ (3)

$$\tilde{\sigma}_{z_s^{[i]}(t)}^2 = \frac{1}{n} \sum_{i=1}^{n} \sigma_{z_s^{[i]}(t)}^{-2} \tilde{\sigma}_{z_s^{[i]}(t)}^2$$ (4)

which is consistent with full GPR ([31, Prop. 2]).

The two summations in (3) and (4) involve the global training dataset. To decentralize the computation, we consider the computation of the two summations as a dynamic average consensus problem and use first-order dynamic average consensus (FODAC) in [32] to track the time-varying sums in a distributed manner. Denote $(\theta^{[i]}(t), \xi^{[i]}(t), \lambda^{[i]}(t))$ the consensus states of agent $i$. Each entry of the consensus states, $(\theta^{[i]}(t), \xi^{[i]}(t), \lambda^{[i]}(t))$, estimates $\frac{1}{n} \sum_{i=1}^{n} \sigma_{z_s^{[i]}(t)}^{-2} \tilde{\mu}_{z_s^{[i]}(t)}$, $\frac{1}{n} \sum_{i=1}^{n} \sigma_{z_s^{[i]}(t)}^{-2} \tilde{\sigma}_{z_s^{[i]}(t)}^2$ for each $z_s \in \mathcal{Z}_{agg}$. State $\lambda^{[i]}(t)$ is used as one of the criteria for applying fusion between agent-based GPR and distributed GPR in fused GPR. The dynamics of FODAC is shown in Lines 4, 6, and 8 of distributed GPR, respectively, for each consensus state, where $\Delta r(t)$ denotes the temporal change of the signal $r$. Specifically, $\theta^{[i]}(t)$ tracks the average of the signal $r_{\theta}(t)$ defined in Line 3 among the agents. In particular, agent $i$ computes a convex combination of $\theta^{[j]}(t)$ for $j \in \{i\} \cup \mathcal{N}(t-1)$, and then, adds the combination into the temporal change of $r_{\theta}(t)$. The update laws for $\xi^{[i]}(t)$ and $\lambda^{[i]}(t)$ are similar. The updated states are sent to each agent in $\mathcal{N}(t)$ as in Line 13. Notice that consensus is not necessarily reached at each time $t$. We will show that consensus is reached in an asymptotic way in Section V-B.

C. Fused GPR

Fused GPR aims to refine predictions of $\eta(z_s)$ by integrating agent-based GPR with distributed GPR. The goal is to obtain an estimate of the predictive distribution $P[\eta(z_s)|\mathcal{D}(t)]$ for each $z_s \in \mathcal{Z}_{agg}$. Note that distributed GPR obtains new estimates
of $\eta(Z_{\text{agg}})$ by combining results from each agent through convex combination. It can return results with more uncertain predictions, and these predictions should be ignored. The set of inputs predicted by distributed GPR with lower uncertainty is defined as $Z_{\text{agg}}[i](t)$ (Line 2 of fused GPR). Set $Z_{\text{agg}}[i](t)$ is the set of inputs $z_{\text{agg}} \in Z_{\text{agg}}$, where the two variance estimates from distributed GPR, the estimates of $\hat{\sigma}^2_{Z_{\text{agg}}[i](t)}$ in (4) and the estimates of $\frac{1}{n} \sum_{i=1}^{n} \hat{\sigma}^2_{Z_{\text{agg}}[i](t)}$, are lower than $\hat{\sigma}^2_{Z_{\text{agg}}[i](t)}$ from agent-based GPR. If this set is empty (Lines 3 and 4 in fused GPR), the results from distributed GPR are ignored and those from agent-based GPR are used. Otherwise, $P\{\eta(z_s), D(t)\}$ is estimated as follows.

Notice that for all $z_{\text{agg}} \in Z_{\text{agg}}[i](t)$, we have

$$P\{\eta(z_s), D(t)\} = \int P\{\eta(z_{\text{agg}}), \eta(z_s)\} d\eta(z_{\text{agg}}).$$

However, $P\{\eta(z_{\text{agg}}), D(t)\}$ and $P\{\eta(z_s), \eta(z_{\text{agg}}), D(t)\}$ are unknown but can be estimated using the results from distributed GPR and agent-based GPR, respectively. In particular, the results from distributed GPR are used to estimate $P\{\eta(z_{\text{agg}}), D(t)\}$ since the estimate has lower variance (uncertainty). The results from agent-based GPR are used to estimate $P\{\eta(z_s), \eta(z_{\text{agg}}), D(t)\}$ because distributed GPR is limited to $Z_{\text{agg}}$ and $z_s$ may not be in $Z_{\text{agg}}$. The product of $P\{\eta(z_s), \eta(z_{\text{agg}}), D(t)\}$ and $P\{\eta(z_{\text{agg}}), D(t)\}$, which yields $P\{\eta(z_s), \eta(z_{\text{agg}}), D(t)\}$, then contains information from the local agent and those from the other agents in the network. The overall process can be interpreted as a fusion of global information with local information, where improvement is expected because more information is provided. After integrating over $\eta(z_{\text{agg}})$, we obtain the estimate of $P\{\eta(z_s), D(t)\}$. The detailed procedure is broken down into the following steps.

**Step 1: Estimation of $P\{\eta(z_{\text{agg}}), D(t)\}$**—Consider any $z_{\text{agg}} \in Z_{\text{agg}}$. Agent $i$'s estimate of $P\{\eta(z_{\text{agg}}), D(t)\}$, denoted by $\tilde{P}^i\{\eta(z_{\text{agg}}), D(t)\}$, is given by distributed GPR.

**Step 2: Estimation of $P\{\eta(z_s), \eta(z_{\text{agg}}), D(t)\}$**—Note that agent-based GPR does not return covariance $\text{cov}(\eta(z_s), \eta(z_{\text{agg}}))$ that reflects the correlation between $\eta(z_{\text{agg}})$ and $\eta(z_s)$ similar to $k(z_s, z_{\text{agg}})$. We set $\text{cov}(\eta(z_s), \eta(z_{\text{agg}})) = g(z_s, t)k(z_s, z_{\text{agg}})$ and define

$$g(z_s, t) \triangleq \min \left\{ \hat{\sigma}^2_{Z_{\text{agg}}[i](t)}, \hat{\sigma}^2_{\eta_{\text{agg}}[i](t)}, \hat{\sigma}^2_{\eta_{\text{agg}}[i](t)} \right\} \cdot \max \{0, c - \psi[i]\} / k(z_s, z_s),$$

where $c \triangleq \hat{\mu}_X^{-1} \left\{ \frac{1}{n} \sum_{i=1}^{n} \chi[i] \psi[i] \right\}$, $\psi[i] \triangleq \frac{\sigma^2}{\sigma^2 + \sigma^2_i}$, $\chi[i] \triangleq \frac{1}{(\alpha^2)^2} + \frac{1}{\sigma^2}$, and $\hat{\mu}_X \triangleq \frac{1}{n} \sum_{i=1}^{n} \chi[i]$. A distributed method for agent $i$ to obtain $\sigma^2[i]$, $j \neq i$, is given in Section IV-D. This ensures that covariance matrix

$$\tilde{\Sigma}_{z_s, z_{\text{agg}}}[i](t) \triangleq \begin{bmatrix} \hat{\sigma}^2_{Z_{\text{agg}}[i](t)} & g(z_s, t)k(z_s, z_{\text{agg}}) & \hat{\sigma}^2_{\eta_{\text{agg}}[i](t)} \\ g(z_s, t)k(z_s, z_{\text{agg}}) & \hat{\sigma}^2_{\eta_{\text{agg}}[i](t)} & \hat{\sigma}^2_{\eta_{\text{agg}}[i](t)} \end{bmatrix}$$

is positive definite. We further verify that this choice is valid by showing $(\hat{\sigma}^2_{Z_{\text{agg}}[i](t)})^2 > 0$ for all $t \geq 1$ and $z_s \in Z_s$ in Section V-B3. We can write

$$\tilde{P}^i\{\eta(z_s), \eta(z_{\text{agg}}), D(t)\} \triangleq \mathcal{N} \left( \begin{bmatrix} \hat{\mu}_{z_s}[i](t) \\ \hat{\mu}_{z_{\text{agg}}}[i](t) \end{bmatrix}, \tilde{\Sigma}_{z_s, z_{\text{agg}}}[i](t) \right).$$

Then, agent $i$'s estimate of $P\{\eta(z_s), \eta(z_{\text{agg}}), D(t)\}$, denoted by $\tilde{P}^i\{\eta(z_s), \eta(z_{\text{agg}}), D(t)\}$, is given by $\tilde{P}^i\{\eta(z_s), \eta(z_{\text{agg}}), D(t)\}$ applying identities of joint Gaussian distribution (16, p. 200) on $P\{\eta(z_s), \eta(z_{\text{agg}}), D(t)\}$.

**Step 3: Estimation of $P\{\eta(z_s), D(t)\}$**—Combining the previous two steps, agent $i$ estimates $P\{\eta(z_s), \eta(z_{\text{agg}}), D(t)\}$ as

$$\tilde{P}^i\{\eta(z_s), \eta(z_{\text{agg}}), D(t)\} = \tilde{P}^i\{\eta(z_s), D(t)\} \cdot \tilde{P}^i\{\eta(z_s), \eta(z_{\text{agg}}), D(t)\}.$$

Applying the same trick of nearest-neighbor prediction as in agent-based GPR, we choose $z_{\text{agg}} = z_{\text{agg}}[i](t) \in \text{proj}(z_s, z_{\text{agg}}[i](t))$ for each $z_s \in Z_s$. Then, we have agent $i$'s estimate of $P\{\eta(z_s), D(t)\}$ given by

$$\tilde{P}^i\{\eta(z_s), D(t)\} = \int \tilde{P}^i\{\eta(z_{\text{agg}}[i](t)), \eta(z_s)|D(t)\} d\eta(z_{\text{agg}}[i](t))$$

which has mean and variance in Lines 11 and 12 of fused GPR (see Section V-A for derivation).
D. Choice of the Kernel

In this article, we assume the following properties of the kernel $k$ used in LiDGPR algorithm.

Assumption IV.1:

1) (Decomposition): The kernel function $k(\cdot, \cdot)$ can be decomposed in such a way that $k(\cdot, \cdot) = \kappa(\rho(\cdot, \cdot))$, where $\kappa : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is continuous.

2) (Boundedness): It holds that $0 < \kappa(r) \leq \sigma_f^2$ for all $r > 0$ and some $\sigma_f > 0$.

3) (Monotonicity): It holds that $\kappa(r)$ is monotonically decreasing as $r$ increases and $\kappa(0) = \sigma_f^2$.

Remark IV.2: In GPR, kernel can be interpreted as the prior correlation between function evaluations. For a continuous function, it is reasonable to assume bounded correlation and the correlation is negatively related to the distance between two inputs. One example that satisfies Assumption IV.1 above is the class of squared exponential kernels having the form $k(z, z') = \sigma_f^2 \exp(-||z - z'||^2/\ell^2)$ ([6], p. 83)).

To obtain the theoretical guarantees in Section IV-E, $\sigma_f^2$ is chosen for initialization as follows. Let $\sigma_f^2 = \sum_{i=1}^n (\gamma_i^0 + \mu_i)^2$, $\gamma_i^0 \triangleq \{ \gamma_i^0, \psi_i^0 \}$, $\psi_i^0 \triangleq \min_{z \in \mathcal{V}} \{ \epsilon_i^2 + ||z - z_i^0|| \}$. We choose $\sigma_f^2 \geq 1$ satisfying

\[
\sigma_f^2/(\mu_i^2 \epsilon_i^2) \leq (\sigma_e^{\text{min}})^2/(\sigma_e^{\text{max}})^2. \tag{5}
\]

When $\sigma_f^2$ increases, $\mu_i^2$, $\mu_i^2$, and $\sigma_f^2$ converge to positive constants, $\epsilon_i^2$ has growth rate $O(\sigma_f^2)$, which gives the left-hand side of (5) diminishing at $O\left(\left(\frac{1}{\sigma_f^2}\right)\right)$. Hence, inequality (5) is satisfied when $\sigma_f^2$ is sufficiently large.

A distributed way to choose a single $\sigma_f^2$ is as follows. By using the Floodset algorithm ([33], p. 103)), each agent $i$ sends $\sigma_i^2$ to its neighbors. By Assumption II.1, within $n(b-1)$ iterations, each agent obtains a copy of $\sigma_f^2$ from all $i \in \mathcal{V}$. Then, the values in (5) can be calculated. To further consider data fitting, each agent can incorporate (5) with existing hyperparameter optimization methods, such as [34], which uses a given amount of data points collected during initialization, or [35], which recursively updates when new data arrive. The resulting local hyperparameter of agent $i$ is denoted as $\sigma_i^2$, then all the agents employ maximum consensus [36] to compute $\sigma_f = \max_{i \in \mathcal{V}} \{ \sigma_i^2 \}$, which terminates in $n(b-1)$ iterations.

E. Performance Guarantee

In this section, we present the performance of predictive mean and variance returned by LiDGPR. The main results are summarized in Theorems IV.3 and IV.8, and their proofs are presented in Sections V-B and V-C.

Part of the performance is quantified in terms of the dispersion of local data defined as $d_i(t) \triangleq \sup_{z \in \mathcal{Z}} \rho(z, Z_i(t))$. We can interpret dispersion as a measurement of how dense the sampled data are distributed within a compact space. For notational simplicity, we introduce shorthand $\rho_z^2 \triangleq \rho(z, Z)$. Theorem IV.3 shows that LiDGPR makes predictions with lower uncertainty than agent-based GPR.

Theorem IV.3. (Uncertainty reduction): Part I: Suppose Assumption IV.1 holds. For all $z_i \in \mathcal{Z}$ and $i \in \mathcal{V}$, the predictive variance by agent-based GPR is bounded as

\[
\frac{\sigma_f^2 (\sigma_e^2)^2}{\sigma_f^2 + (\sigma_e^2)^2} \leq \hat{\sigma}^2_{z_i, [d_i]}(t) \leq \sigma_f^2 - \frac{\kappa(d_i^2(t))}{\sigma_f^2 + (\sigma_e^2)^2}.
\]

Part II: Suppose Assumptions II.1–II.3 and IV.1 hold. For all $z_i \in \mathcal{Z}$ and $i \in \mathcal{V}$, there exists a nonnegative sequence $\gamma_i^0, z_i$, such that the predictive variance by LiDGPR is

\[
0 < (\hat{\sigma}^2_{z_i, [d_i]}(t)) = \sigma_f^2 - \gamma_i^0, z_i, (t).
\]

In particular, if $Z_{agg}^i(t) = \emptyset, \gamma_i^0, z_i, (t) = 0$; otherwise

\[
\gamma_i^0, z_i, (t) \geq \mathcal{O}\left(\kappa(\rho(Z_i, Z_{agg}^i(t)))^2 \left(\frac{1}{n} \sum_{j=1}^n \kappa(\rho(Z_{agg}^i, Z_{agg}^j(t)))^2 \right) - \frac{\sigma_f^2 + (\sigma_e^2)^2}{\sigma_f^2 + (\sigma_e^2)^2} \kappa(\rho(Z_i, Z_{agg}^i(t)))^2 \right).
\]

We provide the steady-state results assuming that the dispersion is diminishing. [37, Lemma 6] shows that dispersion does go to zero under uniform sampling.

Corollary IV.4: If $\lim_{t \to \infty} d_i(t) = 0$ for all $i \in \mathcal{V}$ and all the conditions in Theorem IV.3 are satisfied, then

\[
\lim \inf_{t \to \infty} \gamma_i^0, z_i, (t) \geq \mathcal{O}\left(\lim \sup_{t \to \infty} \sigma_f^2_{z_i, [d_i]}(t) - \frac{1}{n} \sum_{j=1}^n \lim \sup_{t \to \infty} \sigma_f^2_{z_i, [d_i]}(t) \right)
\]

where $\lim_{t \to \infty} \sigma_f^2_{z_i, [d_i]}(t) = \sigma_f^2 (\sigma_e^2)^2/(\sigma_f^2 + (\sigma_e^2)^2)$. To ensure the improvement on prediction accuracy, we need to assume that the prior covariance function of $\eta$ is correctly specified. Note that any nonzero mean Gaussian process can be decomposed into a deterministic process plus a zero-mean stochastic process such that GPR can be performed over the zero-mean stochastic process ([6], p. 27)). Therefore, without loss of generality, we assume $\eta$ follows a zero-mean Gaussian process for notational simplicity.

Assumption IV.5: It satisfies that $\eta \sim G\mathcal{F}(0, k)$. That is, the target function $\eta$ is completely specified by a zero-mean Gaussian process with kernel $k$. This assumption is common in the analysis of GPR ([38], Th. 1)).

Furthermore, we need to assume that the state transition matrix induced by $A(t)$ is constant.

Assumption IV.6: It holds that $\prod_{t=1}^T A(\tau) = \prod_{t=1}^T A(\tau)$ for any $t, \tau > 1$.

One example that satisfies this assumption is each entry of $A(t)$ being constant $\alpha$, which is a complete graph. Furthermore, we assume $\eta$ is Lipschitz continuous.

Assumption IV.7: There exists some positive constant $\bar{\epsilon} \in \mathbb{R}$ such that $\sup_{z \in \mathcal{Z}} \| \eta(z) - \eta(z') \| \leq \bar{\epsilon} \rho(z, z')$.

Theorem IV.8 below compares the predictive errors of agent-based GPR with those of LiDGPR.

Theorem IV.8. (Accuracy improvement): Part I: Suppose Assumptions IV.1, IV.5, and IV.7 hold. For all $z_i \in \mathcal{Z}$ and $i \in \mathcal{V}$, with probability at least $1 - (\omega^{\text{max}})^2$, where $\epsilon > \sigma_e^{\text{max}}$, the predictive error resulted from agent-based GPR is bounded as

\[
\mathcal{E}_\eta|_{z_i, [d_i]}(t) = \| \mu(z_i, [d_i])(t) - \eta(z_i) \| \leq \left(1 - \frac{\kappa(d_i^2(t))}{\sigma_f^2 + (\sigma_e^2)^2}\right) \| \eta \| \| z_i \| + \bar{\epsilon} \rho(z_i, z_i^0) + \epsilon.
\]
Part II: Suppose \( \lim_{t \to \infty} d[i](t) = 0 \) \( \forall i \in V \), and Assumptions II.1–II.3, IV.1, and IV.5–IV.7 hold. For all \( z_i \in Z \) and \( i \in V \); if \( \lim_{t \to \infty} Z_{agg}^i(t) \neq \emptyset \), then

\[
\lim_{t \to \infty} \mathbb{E} \left[ \left( \hat{\mu}_{zi}[D(t)] - \eta(z_i) \right)^2 \right] = \left( \hat{\mu}_{zi}[D](t) - \eta(z_i) \right)^2 \leq -O \left( \kappa(\rho_{agg}) \right) = 0
\]

otherwise, \( \lim_{t \to \infty} \mathbb{E} \left[ \left( \hat{\mu}_{zi}[D(t)] - \eta(z_i) \right)^2 \right] = \lim_{t \to \infty} \left( \hat{\mu}_{zi}[D](t) - \eta(z_i) \right)^2 \). Further, if \( \sigma_e[i]^2 > \frac{1}{n} \sum_{j=1}^{n} (\sigma_e[j])^2 \), \( \lim_{t \to \infty} Z_{agg}^i(t) = Z_{agg} \).

The two theorems indicate that LiDGPR leverages interagent communication to improve transient and steady-state learning performances; meanwhile, no agent suffers from degraded learning performance. This improvement of learning performance is achieved by the fact that the agents whose data samples have higher dispersion (or observation noise has higher variance) benefit from those with data samples having lower dispersion (or observation noise having lower variance) via communication. Next we elaborate on the fact.

**Transient improvement:** Term \( \frac{1}{n} \sum_{j=1}^{n} \kappa(\rho_{agg})^2 \) are closely related to dispersion \( d[i](t) \) and recall the monotonicity property of \( \kappa \). Hence, LiDGPR enables agents whose data samples have higher dispersion (data sparsely sampled) and observation noise has higher variance to benefit from those with data samples having lower dispersion (data densely sampled) and observation noise having lower variance.

**Steady-state improvement:** If \( \lim_{t \to \infty} \gamma[\sigma_{agg}](t) > 0 \), it indicates that agent \( i \) obtains improvement in steady-state learning performance in predictive variance. From Corollary IV.4, we can see that \( \lim_{t \to \infty} \gamma[\sigma_{agg}](t) > 0 \) if its steady-state local predictive variance, \( \lim_{t \to \infty} \sigma_{agg}^2[D](t) \), is above the average over the agents in \( V \). By Corollary IV.4, \( \lim_{t \to \infty} \sigma_{agg}^2[D](t) \) is positively related to \( (\sigma_e[i])^2 \). Hence, agents with observation noise of higher variance might obtain steady-state improvement in predictive variance from those with lower variance.

**Steady-state improvement in prediction accuracy** is reflected by the case \( \lim_{t \to \infty} Z_{agg}^i(t) \neq \emptyset \) in Theorem IV.8. The sufficient condition \( (\sigma_e[i])^2 > \frac{1}{n} \sum_{j=1}^{n} (\sigma_e[j])^2 \) indicates that agent \( i \) obtains steady-state improvement when \( (\sigma_e[i])^2 \) is above the average. That is, agents with observation noise of higher variance benefits from those with smaller variance.

The improvements \( \gamma[\sigma_{agg}](t) \) and \( \lim_{t \to \infty} \mathbb{E} \left[ (\hat{\mu}_{zi}[D(t)] - \eta(z_i))^2 \right] = (\hat{\mu}_{zi}[D](t) - \eta(z_i))^2 \) are positively related to \( \kappa(\rho_{agg})^2 \) and \( \kappa(\rho_{agg})^2 \), respectively. By monotonicity of \( \kappa \) in Assumption IV.1, these terms indicate the benefit brought by communication decays as \( z_i \) is moving away from \( z_{agg} \) and \( Z_{agg} \), respectively. That is, a denser set \( Z_{agg} \) could induce larger improvements.

**F. Discussion**

**Relevance:** The two theorems indicate that both prediction uncertainties and prediction errors reduce as local dispersion \( d[i](t) \) reduces. This provides insights on data sampling such that the agents should sample in a way that minimizes \( d[i](t) \). The terms \( \kappa(\rho_{agg})^2 \) and \( \kappa(\rho_{agg})^2 \) in Theorems IV.3 and IV.8 show that the improvement of learning performances obtained from communication decreases as the test point \( z_i \) is moving away from \( Z_{agg} \). This can guide the design process of \( Z_{agg} \) such that if the test points in \( Z_i \) are known a priori, \( Z_{agg} \) should be allocated such that \( \sup_{z_i, \in Z_i} \min_{\rho_{agg}} |D(\eta(z_i), \rho(z_i), Z_{agg})| \) is minimized; otherwise, \( Z_{agg} \) should be designed such that \( \sup_{z_i, \in Z_i} \min_{\rho_{agg}} |D(\eta(z_i), \rho(z_i), Z_{agg})| \) is minimized.

**Complexities related to \( Z_i \) and \( Z_{agg} \):** The communication overhead scales as \( O(|Z_{agg}| N^{(i)}(t)) \). Due to the use of nearest-neighbor GPR, agent-based GPR only requires \( O(t) \) in memory. The memory requirements for both distributed GPR and fused GPR are \( O(|Z_{agg}|) \). The computational complexities scale as \( O(t|Z_i|) \) for agent-based GPR, \( O(|Z_{agg}|) \) for distributed GPR, and \( O(|Z_i| |Z_{agg}|) \) for fused GPR.

**Nearest-neighbor GPR versus full GPR:** Part I of Theorems IV.3 and IV.8 characterizes the steady-state errors of agent-based GPR. Choi and Schervish [7] showed that \( \sigma^2[D](t) \) and \( \rho_{agg}(z_{agg}) \) for nearest-neighbor GPR have the same steady-state errors under \( \gamma[\sigma_{agg}](t) \). Hence, the variance for noisy prediction \( (\sigma_e[i])^2 \), and \( \sigma_{agg}^2[D](t) \) in Theorem IV.8 indicates

\[
\limsup_{t \to \infty} \|D(\gamma[\sigma_{agg}](t) - \eta(z_i)) \|_{Z_{agg}} = \left( \sigma_e[i] \right)^2 + \left( \sigma_{agg} \right)^2 \|\eta\|_{Z} + \epsilon
\]

assuming \( d[i](t) \) to be zero. The discrepancy can be caused by the fact that full GPR in [7] makes prediction using all the data in the dataset whereas nearest-neighbor GPR only uses the data of the nearest input. Full GPR has computational complexity \( O(t^5) \) while nearest-neighbor GPR has the same computational complexity as nearest-neighbor search, which is \( O(t) \) for the worst case [39]. This is the tradeoff between learning accuracy and computational complexity. Note that both full GPR and nearest-neighbor GPR have the same steady-state errors under noise-free condition, i.e., \( (\sigma_e[i])^2 = 0 \).

**V. PROOFS**

In this section, we present the derivation of Lines 11 and 12 in fused GPR and the proofs of Theorems IV.3 and IV.8. Table I gives the symbols that are used in multiple important results and the relation among them.

**A. Derivation of Lines 11 and 12 in Fused GPR**

Recall that \( \tilde{P}^i_t \{ \eta(z_i) | D[i](t) \} \) is given by applying identities of joint Gaussian distribution (6, p. 200) to

\[
\tilde{P}^i_t \left\{ \eta(z_i) \left| \begin{array}{c} \eta(z_{agg}) \end{array} \right. \right\} \| D[i](t) \rangle = N \left( \begin{array}{c} \tilde{\mu}_{zi}[D](t) \\ \tilde{\mu}_{agg}[D](t) \end{array} \right), \tilde{\Sigma}_{z_i, z_{agg}[D](t) \rangle}
\]

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
This gives \( \tilde{P}^{[i]} \{ \eta(z_s) | \eta(z_{agg}) , D(t) \} \) as a Gaussian distribution with mean and variance
\[
\tilde{\mu}^{[i]}_{z_s | z_{agg}, D} \left( t \right) = \mu_{z_s | \mu_{z_{agg}}, D} ( t ) + g(z_s, t) k(z_s, z_{agg}) \sigma^2_{z_{agg} | D} ( t ) \eta(z_{agg}) - \mu_{z_{agg} | D} ( t ) \\
\tilde{\sigma}^2_{z_s | z_{agg}, D} ( t ) = \sigma^2_{z_s | D} ( t ) + g(z_s, t) k(z_s, z_{agg}) \sigma^2_{z_{agg} | D} ( t ) k(z_s, z_{agg}) g(z_s, t).
\]

Notice that the mean and variance of \( \tilde{P}^{[i]} \{ \eta(z_s) | \eta(z_{agg}) , D(t) \} \) are given in distributed GPR. Then, we have the product
\[
\tilde{P}^{[i]} \{ \eta(z_s) , \eta(z_{agg}) | D(t) \} = \tilde{P}^{[i]} \{ \eta(z_s) | \eta(z_{agg}) , D(t) \} \cdot \tilde{P}^{[i]} \{ \eta(z_s) , \eta(z_{agg}) | D(t) \}
= \mathcal{N} \left( \tilde{\mu}^{[i]}_{z_s | z_{agg}, D} ( t ) , ( \tilde{\sigma}^2_{z_s | z_{agg}, D} ( t ) ) \right) \cdot \mathcal{N} \left( \tilde{\mu}^{[i]}_{z_s | D} ( t ) , ( \tilde{\sigma}^2_{z_s | D} ( t ) ) \right)
= \mathcal{N} \left( \begin{bmatrix} \tilde{\mu}^{[i]}_{z_s | D} ( t ) \\ \tilde{\mu}^{[i]}_{z_s | z_{agg}, D} ( t ) \end{bmatrix} , \begin{bmatrix} \tilde{\sigma}^2_{z_s | D} ( t ) & \tilde{\sigma}^2_{z_s | z_{agg}, D} ( t ) \\ \tilde{\sigma}^2_{z_s | z_{agg}, D} ( t ) & \tilde{\sigma}^2_{z_s | z_{agg} | D} ( t ) \end{bmatrix} \right) .
\]

Replacing \( z_{agg} \) with \( \tilde{z}_{agg}^{[i]} ( t ) , \tilde{\mu}^{[i]}_{z_s | z_{agg}, D} ( t ) \) and \( ( \tilde{\sigma}^2_{z_s | z_{agg}, D} ( t ) )^2 \) have the forms in Lines 11 and 12 in fused GPR. Hence, we have the marginal distribution \( \tilde{P}^{[i]} \{ \eta(z_s) | D(t) \} \).

### B. Proof of Theorem IV.3

In this section, we first derive the lower bound and the upper bound of the predictive variance of agent-based GPR and verify Part I of Theorem IV.3 in Section V-B1. Then, we derive the bounds of distributed GPR in Proposition V.4 in Section V-B2. Finally, we derive the bounds of fused GPR and prove Part II of Theorem IV.3 in Section V-B3.

First of all, we introduce some properties of functions \( f_1 : \mathbb{R}_{>0} \rightarrow \mathbb{R}_{>0} \) as \( f_1 ( x ) = x \) and \( f_2 : \mathbb{R}_{>0} \rightarrow \mathbb{R}_{>0} \) as \( f_2 ( x ) = x ( 1 + \frac{\sigma^2}{\sigma^2_f} ) \). These will be used in later analysis.

**Lemma VI.1:** It holds that
\[
f_1 \left( \frac{1}{n} \sum_{i=1}^{n} x_i \right) \leq \frac{1}{n} \sum_{i=1}^{n} f_1 ( x_i ) \leq \frac{1}{n} \sum_{i=1}^{n} f_2 ( x_i ) \leq f_2 \left( \frac{1}{n} \sum_{i=1}^{n} x_i \right)
\]
\[
\left( \frac{1}{n} \sum_{i=1}^{n} f_2(x_i) \right)^{-1} \leq \frac{1}{n} \sum_{i=1}^{n} (f_2(x_i))^{-1}.
\]

**Proof:** Due to space limitation, the proof is included in the complete version [25].

1) **Variance Analysis of Agent-Based GPR:** In this section, we present the proof of Theorem IV.3 Part I.

**Proof of Theorem IV.3 Part I:** Pick any \( z_0 \in \mathcal{Z} \). By monotonicity of \( \kappa \) in Assumption IV.1, Line 5 in agent-based GPR gives the predictive variance \( \hat{\sigma}^2_{g, |D|}(t) = \sigma^2_f - \frac{k(z_0, z_{[i]}(t))^2}{\sigma^2_f + \sigma^e_i \hat{\sigma}^2_{g, |D|}(t)} \).

Note that the definition of \( z_{[i]}(t) \) renders \( \hat{\rho}_{z_0} = \hat{\rho}(z_{[i]}(t)). \)

Combining this with the decomposition property of \( \kappa \) in Assumption IV.1 gives

\[
\hat{\sigma}^2_{z, |D|}(t) = \sigma^2_f - \frac{\kappa(\hat{\rho}_{z_0})^2}{\sigma^2_f + \sigma^e_i \hat{\sigma}^2_{g, |D|}(t)}.
\]

The definition of local dispersion \( \hat{d}^{[i]}(t) \) renders \( \hat{d}^{[i]}(t) \geq \hat{\rho}(z_{[i]}, \mathcal{Z}(t)) \).

Combining this with the monotonicity of \( \kappa \) in Assumption IV.1 gives \( \kappa(\hat{\rho}_{z_0}) \leq \kappa(\hat{\rho}(z_{[i]}(t))) \), which renders \( \hat{\sigma}^2_{z, |D|}(t) \leq \hat{\sigma}^2_f - \frac{\hat{d}^{[i]}(t)^2}{\sigma^2_f + \sigma^e_i \hat{\sigma}^2_{g, |D|}(t)} \) \( \forall z_0 \in \mathcal{Z} \).

Applying the boundedness of \( \kappa \) in Assumption IV.1 to (6), we have \( \hat{\sigma}^2_{z, |D|}(t) \geq \frac{\sigma^2_f(\sigma^e_i)^2}{\sigma^2_f + \sigma^e_i \hat{\sigma}^2_{g, |D|}(t)} \).

As \( \lim_{t \to \infty} \hat{d}^{[i]}(t) = 0 \) \( \forall j \in V \), the upper bound of \( \hat{\sigma}^2_{z, |D|}(t) \) converges to its lower bound.

**Corollary V.2:** Suppose Assumption IV.1 holds. If \( \lim_{t \to \infty} \hat{d}^{[i]}(t) = 0 \) \( \forall j \in V \), it holds that \( \lim_{t \to \infty} \hat{\sigma}^2_{z, |D|}(t) = \frac{\sigma^2_f(\sigma^e_i)^2}{\sigma^2_f + \sigma^e_i \hat{\sigma}^2_{g, |D|}(t)} \).

2) **Variance Analysis of Distributed GPR:** First, we define the following notations. We let operators \( \Delta, \sup, \min \) and min be applied elementwise across the vectors

\[
\hat{m}_i(t) = \max_{i \in V} \xi_i^{[i]}(t), \quad \underline{m}_i(t) = \min_{i \in V} \xi_i^{[i]}(t)
\]

\[
\delta_m(t) \triangleq \hat{m}_i(t) - \underline{m}_i(t), \quad \delta_r^{[i]}(t) \triangleq \max_{i \in V} \delta_r^{[i]}(t), \quad \delta_{r_{min}}(t) \triangleq \min_{i \in V} \delta_r^{[i]}(t)
\]

\[
\Delta r_{\max}(t) \triangleq \max_{i \in V} \Delta r^{[i]}(t), \quad \Delta r_{\min}(t) \triangleq \min_{i \in V} \Delta r^{[i]}(t)
\]

\[
\hat{\delta}_r(t) \triangleq \Delta r_{\max}(t) - \Delta r_{\min}(t), \quad \hat{\zeta}(t) \triangleq \frac{1}{n} \sum_{i=1}^{n} \xi_i^{[i]}(t) - \delta r^{[i]}(t - 1)
\]

First of all, we introduce several properties in Lemma V.3.

**Lemma V.3:** Suppose Assumptions II.2 and IV.1 and \( \lim_{i \to \infty} \hat{d}^{[i]}(t) = 0 \) \( \forall j \in V \) hold. For each \( z_0 \in \mathcal{Z}_{agg} \) the following holds.

**Claim V.3.1.** It holds that \( \frac{1}{\sigma^2_f} \leq \frac{\hat{\delta}_r(t)}{\hat{\zeta}^{[i]}(t)} \leq n \frac{\sigma^2_f(\sigma^e_i)^2}{\sigma^2_f + \sigma^e_i \hat{\sigma}^2_{g, |D|}(t)} \) \( \forall i \in V, t \geq 0 \).

**Claim V.3.2.** It holds that \( \hat{\xi}^{[i]}(t) \geq \hat{\xi}^{[i]}(t - 1) \) \( \forall t \geq 0 \).

**Claim V.3.3.** It holds that \( \frac{1}{\sigma^2_f} \leq \frac{\hat{\delta}_r(t)}{\hat{\zeta}^{[i]}(t)} \leq n \frac{\sigma^2_f(\sigma^e_i)^2}{\sigma^2_f + \sigma^e_i \hat{\sigma}^2_{g, |D|}(t)} \) \( \forall i \in V, t \geq 0 \).

Claim V.3.4. It holds that \( \hat{\delta}_r(t) \leq \Delta r_{\max}(t) \) \( \forall t \geq 0 \).

Claim V.3.5. It holds that \( \Delta r_{\max}(t) \leq \Delta r_{\max}(t - 1) \) \( \forall t \geq 1 \).

**Proof:** Due to space limitation, the proof is included in the complete version [25].

We define the subsequence \( \{t_j\} \) as follows: \( t_{-1} \triangleq 1 \)

\[
t_0 \triangleq \left( \log \left( \frac{2(n-1)}{\log(1-\zeta)} \right) + 1 \right) \text{ for all } j \geq 1.
\]

The proposition below characterizes the convergence of predictive variance from distributed GPR.

**Proposition V.4.** (Convergence of distributed GPR): Suppose Assumptions II.1–II.3 and IV.1 hold. For all \( z_0 \in \mathcal{Z}_{agg} \), for all \( i \in V \) and \( t \geq 1 \), the convergence of \( \hat{\sigma}^{[i]}(t) \) in distributed GPR to \( \hat{\sigma}^{[i]}_{agg}(t) \) is characterized by: For \( t < t_0 \)

\[
\Delta r_{\max}(t) \leq \frac{4 \sigma_f}{\zeta} \left( \frac{n}{n-2} \right) \Delta r_{\max}(t-1).
\]

For \( t \geq t_0 \)

\[
\Delta r_{\max}(t) \leq \frac{4 \sigma_f}{\zeta} \left( \frac{n}{n-2} \right) \Delta r_{\max}(t-1).
\]

The upper bound of \( \hat{\xi}^{[i]}(t) \) and \( \hat{\xi}^{[i]}_{agg}(t) \) is found by the following two claims, whose proofs are in [25].

**Claim V.4.1.** It holds that \( \Delta r_{\max}(t) \leq \frac{4 \sigma_f}{\zeta} \left( \frac{n}{n-2} \right) \Delta r_{\max}(t-1) \) \( \forall t \geq 0 \).

**Claim V.4.2.** It holds that \( \Delta r_{\max}(t) \leq \frac{4 \sigma_f}{\zeta} \left( \frac{n}{n-2} \right) \Delta r_{\max}(t-1) \) \( \forall t \geq 0 \).

Claim V.3.1 and Claim V.3.3 in Lemma V.3 provide that \( \hat{\xi}^{[i]}(t) \geq \frac{1}{\sigma_f} \) and \( \hat{\xi}^{[i]}_{agg}(t) \geq \frac{1}{\sigma_f} \), respectively. Combining this and Claim V.4.1 and Claim V.4.2 with (7) finishes the proof.
Corollary V.5 shows that the predictive variance from distributed GPR converges to that from the aggregated method.

**Corollary V.5**: Suppose the same conditions as in Proposition V.4 hold. If \( \lim_{t \to \infty} d[i](t) = 0 \) for all \( i \in \mathcal{V} \), then \( \lim_{t \to \infty} Z_{\text{agg}}[i](t) = 0 \).

**3) Variance Analysis of Fused GPR**: First of all, we show that \( \lim_{t \to \infty} Z_{\text{agg}}[i](t) \) exists.

**Lemma V.6**: It holds that \( \lim_{t \to \infty} Z_{\text{agg}}[i](t) \) exists.

**Proof**: Due to space limitation, the proof is included in the complete version [25].

**Lemma V.7**: Suppose the same conditions for Corollary V.5 hold and \( d[i](t) \to 0 \) \( \forall j \in \mathcal{V} \). If \( \lim_{t \to \infty} Z_{\text{agg}}[i](t) \neq 0 \) for some \( i \in \mathcal{V} \), then \( \psi[i] < \frac{1}{n} \sum_{j=1}^{n} \psi[j] \leq c \) and \( \mu_{\tilde{\gamma}[i]} < 1 \).

**Proof**: Due to space limitation, the proof is included in the complete version [25].

Now, we present the proof of Theorem IV.3 Part II.

**Proof of Theorem IV.3 Part II**: By Line 4 in fused GPR, it is obvious that if \( Z_{\text{agg}}[i](t) = 0 \), \( \gamma[i,z,t] = 0 \). Now, we consider the case when \( Z_{\text{agg}}[i](t) \neq 0 \).

**Outline**: The proof is composed of three parts: expression of \( \gamma[i,z,t] \) and its uniform lower bound, verification of the selection of \( g(z_i,t) \), and derivation of the growth factor of \( \gamma[i,z,t] \).

First, we show the expression of \( \gamma[i,z,t] \) and derive its uniform lower bound. According to Line 12 in fused GPR, we have

\[
\gamma[i,z,t] = \gamma[i,z,1,t] \gamma[i,z,2,t] \gamma[i,z,3,t] \left( \gamma[i,\min,z,t] \right)^2
\]

\[
\left( \gamma[i,\min,z,t] \right)^2 \leq \min \left\{ \sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) |, \sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) | \right\}
\]

\[
\gamma[i,z,1,t] = \frac{k(z_i,z_{\text{agg}}(t))^2 \cdot \max \{0,(c-\psi[i])^2 \}}{k(z_i,z_{\text{agg}}(t))^2}
\]

\[
\gamma[i,z,2,t] = \sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) | - \left( \sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) | \right)^2
\]

\[
\gamma[i,z,3,t] = \frac{\sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) |}{\sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) |}
\]

(8)

Line 2 of fused GPR rules that \( (\sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) |) < (\sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) |) \).

Obviously, \( \gamma[i,z,t] \geq 0 \).

Second, we verify the selection of \( g(z_i,t) \). We verify that the selection of \( g(z_i,t) \) is valid by showing \( (\sigma[i,z,t])^2 > 0 \). We analyze each factor of \( \gamma[i,z,t] \) as follows.

Note that \( c, \psi[i] \in (0,1) \), hence, \( 0 \leq \max \{0,(c-\psi[i])^2 \} \leq 1 \). The decomposition and monotonicity properties in Assumption IV.1 give \( 0 \leq (k(z_i,z_{\text{agg}}(t))^2 \leq 1 \). Combining these gives \( 0 \leq \gamma[i,z,1,t] \leq 1 \).

By definition, \( (\sigma[i,\min,z,t])^2 \leq (\sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) |) \), which renders \( 0 < \gamma[i,z,2,t] < 1 \).

The above upper bounds give \( (\gamma[i,z,t]|D^{[i]}(t)) < (\gamma[i,\min,z,t]|D^{[i]}(t)) \leq (\sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) |) \). By Claim V.3.3 and Line 9 in distributed GPR, \( (\sigma[i,z,t]|D^{[i]}(t)) > 0 \). Therefore, \( 0 < \gamma[i,z,2,t] < 1 \).

The upper bounds in Proposition V.4 can be written as

\[
h(t) \leq \left\{ \begin{array}{ll}
\bar{c}(1 - \bar{c} \xi^{-1} - t, t < t_0 \\
\|\Delta r_{\max}(t)\|_{\infty}, t \geq t_0
\end{array} \right.
\]

(9)

Then, Proposition V.4 gives

\[
\gamma[i,z,t]\gamma[i,z,2,t] = \frac{\sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) | - (\sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) |)^2}{h(t)}.
\]

The upper bound and lower bound of \( (\sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) |) \) is given below, whose proof can be found in [25].

Claim V.7.1. For each \( z_i \in \mathcal{Z} \), the aggregated variance returned from (4) can be characterized as

\[
\sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) | \leq (\sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) |) \leq \left( \frac{\bar{c}}{\bar{c}^2 + (\sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) |)^2} \right).
\]

Denote \( \phi[i] \triangleq \sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) | \) and the upper bound in Claim V.7.1 for \( (\sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) |) \) in the inequality above gives

\[
\gamma[i,z,2,t] \geq \frac{\sigma[i,z,t]}{\phi_{\max}} \left( \frac{\phi[i]}{n} \sum_{j=1}^{n} \kappa(\frac{Z_j[z,t]}{Z_{\text{agg}}[z,t]})^2 \right)^2\left( \frac{\sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) |}{\sigma^2_{\text{agg}} z_{\text{agg}}(t) | D^{[i]}(t) |} - \phi_{\max}(\phi[i] h(t)) \right).
\]
The boundedness of $\kappa$ in Assumption IV.1 gives $|\sigma^2_i \phi_i| - \kappa(\rho_{y_i^0(t)})^2 \leq \sigma^2_i \phi_i$. Applying this upper bound to the denominator of the lower bound above gives

$$
\gamma[t, z, t] (t) = \frac{1}{\phi_i \phi_{\max} \sigma_f} \left( \sum_{j=1}^{n} \kappa \rho_{z_i^0(t)}^2 - \phi_{\max} \phi_i h(t) \right).
$$

Now, we characterize the rest of the factors of $\gamma[t, z, t](t)$. By monotonicity and decomposition in Assumption IV.1, we have $k(z, z) = \sigma_f^2$ and $k(z, z_{\text{agg}}) = (t)^2 = \sigma_f^2$. By Lemma V.7, we have $c - \psi_i \geq \sum_{j=1}^{n} \psi_j \phi_j^2 - \sigma_f^2 > 0$, where $\phi_j^2 \geq 0$ ($\sum_{j=1}^{n} \psi_j \phi_j^2 - \sigma_f^2 > - \phi_i^2$). This gives $\gamma[t, z, t] (t) \geq \kappa(\rho_{y_i^0(t)})^2 \rho_{z_i^0(t)}^2 \phi_i^2 / \sigma_f^2$.

Part I indicates that $\sigma^2_{\gamma[t, z, t], z_i](t)^2 \geq \gamma[t, z, t]^4 \phi_i^2 / \sigma_f^2$.

Equality (6) indicates $\sigma^2_{\gamma[t, z, t], z_i](t)^2 \leq \sigma_f^2$. Combining this with the lower bound of $\gamma[t, z, t](t)^2$ above gives $\gamma[t, z, t](t)^2 \geq \sigma_f^2 / \phi_i^2$.

Combining the lower bounds of all the factors gives

$$
\gamma[t, z, t](t) \geq \kappa(\rho_{y_i^0(t)}^2)^2 \left( \frac{\sigma_f^2}{\phi_i^2} \frac{1}{\phi_{\max} \sigma_f} \sum_{j=1}^{n} \kappa \rho_{z_i^0(t)}^2 - \phi_{\max} \phi_i h(t) \right) \rho_{z_i^0(t)}^2 \phi_i^2 / \sigma_f^2.
$$

The definition in (9) and Claim V.3.6 render $h(t) \rightarrow 0$. This renders the big O notion.

C. Proof of Theorem IV.8

In this section, we present the theoretical results that lead to Theorem IV.8. We first present the error between the predictive mean of agent-based GPR and the ground truth, which is the result of Theorem IV.8 Part I, in Section V-C1. Second, we characterize the predictive mean returned from distributed GPR in Proposition V.9 in Section V-C2. Finally, we finish the proof of Part II of Theorem IV.8 in Section V-C3.

1) **Mean Analysis of Agent-Based GPR:** In this section, we provide the proof of Part I of Theorem IV.8.

Proof of Part I of Theorem IV.8: By Assumption IV.5 and decomposition and monotonicity properties in Assumption IV.1, Line 4 of agent-based GPR becomes $\mu[t, z_i(t) \eta(z)] = \kappa(\rho_{y_i^0(t)}(t) / (\sigma_f^2 + \sigma_i^2))^2$. $y_i^0(t)$ It implies that

$$
\mu[t, z_i(t) \eta(z)] = \left(1 - \frac{\kappa(\rho_{y_i^0(t)}^2)}{\sigma_f^2 + \sigma_i^2} \right)^2 (-\eta(z_i(t))
$$

+ $\kappa(\rho_{y_i^0(t)}^2) / (\sigma_f^2 + \sigma_i^2)^2 \left( y_i^0(t) - \eta(z_i(t)) \right)
$$

+ $\kappa(\rho_{z_i^0(t)}^2) / (\sigma_f^2 + \sigma_i^2)^2 \left( \eta(z_i^0(t)) - \eta(z_i(t)) \right)$.

By boundedness of $\kappa$ in Assumption IV.1, $0 < \kappa(\rho_{y_i^0(t)}^2) / (\sigma_f^2 + \sigma_i^2)^2 < 1$. Combining this with triangular inequality gives

$$
|\mu[t, z_i(t) \eta(z)]| \leq \left(1 - \frac{\kappa(\rho_{y_i^0(t)}^2)}{\sigma_f^2 + \sigma_i^2} \right) ||\eta|| + \|\eta(z_i(t)) - \eta(z_i(t))\|.
$$

Now, we analyze the upper bound of each term on the right-hand side of (10). Recall that $z_i^0(t) \in \text{proj}(z, z_i^0(t))$. Utilizing the Lipschitz continuity of $\eta$ in Assumption IV.7 gives $||\eta(z_i^0(t)) - \eta(z_i(t))|| \leq \epsilon \eta \rho(z, z_i^0(t)) = \epsilon / \rho_{z_i^0(t)}^2$.

The observation model (1) gives $x_i^0(t) \sim \mathcal{N}(\eta(z_i^0(t)), \sigma_i^2)$. Therefore, by Chebyshev inequality (40, p. 151), for all $\epsilon > 0$, we have $P\{|y_i^0(t) - \eta(z_i^0(t))| \geq \epsilon\} \leq \sigma_i^2 / \epsilon^2$. Note that $||\eta(z_i(t))|| \leq \|\eta\|$. Applying these two inequalities to (10) gives

$$
|\mu[t, z_i(t) \eta(z)]| \leq \left(1 - \frac{\kappa(\rho_{y_i^0(t)}^2)}{\sigma_f^2 + \sigma_i^2} \right) \|\eta\| + \epsilon + \epsilon \rho_{z_i^0(t)}^2
$$

with probability at least $1 - \sigma_i^2 / \epsilon^2 \geq 1 - (\sigma_i^2 / \epsilon^2)$.

2) **Mean Analysis of Distributed GPR:** Before presenting the results, we derive the solution to the consensus state $\theta_i^0(t)$, where $i \in \mathcal{V}$, in terms of input signal $\Delta \theta_i^0(t)$. We also show the decompositions of $r_{\theta_i^0(t)}(t)$ and $\theta_i^0(t)$, which separate the two terms into real-valued parts and stochastic parts.

First, we give the solution to $\theta_i^0(t)$. Let vectors $\theta_i^0(t) = \left[ \theta_i^0(t), \ldots, \theta_i^0(n) \right]^T$ and $r_{\theta_i^0(t)}(t) = \left[ r_i^0(t), \ldots, r_i^0(n) \right]^T$. Line 4 of distributed GPR across the network $\mathcal{V}$ can be represented by discrete linear time-varying system: $\theta_i^0(t) = A(t)\theta_i^0(t) - 1) \theta_i^0(t - 1) + \Delta r_{\theta_i^0(t)}(t)$. By [41, p. 111], the solution to this
where $\Phi(t, l) \triangleq \prod_{\tau=l}^{t-1} A(\tau)$.

Second, we show the decomposition of $\Delta r_{\theta_z}(t)$ into a signal depending on $\eta$ and a zero-mean stochastic process. By definition of $r_{\theta_z}(t)$ in Line 3 in distributed GPR and Remark V.8, it holds that

$$r_{\theta_z}(t) = \hat{\delta}_{z, (t)}^2 (\hat{r}_z(t) + \hat{e}_z(t)) = \hat{r}_z(t) + \hat{e}_z(t)$$

where $\hat{r}_z(t) \triangleq \hat{\delta}_{z, (t)}^2 (\hat{r}_z(t))$ and $\hat{e}_z(t) \triangleq \hat{\delta}_{z, (t)}^2 (\hat{e}_z(t))$ is a Gaussian random variable with zero mean. Hence, we have

$$\Delta r_{\theta_z}(t) = \Delta \hat{r}_z(t) + \Delta \hat{e}_z(t).$$

Denote $\hat{r}_z(t) \triangleq \lim_{t \to \infty} \hat{r}_z(t)$. Corollary V.2 and Remark V.8 give $\hat{r}_z(t) = \lim_{t \to \infty} \hat{\delta}_{z, (t)}^2 (\hat{r}_z(t)) = \left( \frac{\sigma^2(\alpha_0^2)^2}{\sigma^2 + (\alpha_0^2)^2} \right) \hat{r}_z(t)$ and

$$\lim_{t \to \infty} \hat{e}_z(t) = \left( \frac{\sigma^2(\alpha_0^2)^2}{\sigma^2 + (\alpha_0^2)^2} \right) \hat{e}_z(t).$$

is zero mean Gaussian.

Third, we show the decomposition of (11). The solution (11) can be decomposed into a solution to FODAC [32] with respect to a signal depending on $\eta$ and a solution to FODAC with respect to a zero-mean stochastic process.

Let $\hat{r}_z(t) \triangleq [\hat{r}_z(1), \ldots, \hat{r}_z(t)]^T$ and $\hat{e}_z(t) \triangleq [\hat{e}_z(1), \ldots, \hat{e}_z(t)]^T$. By (12), we can write (11) as

$$\theta_z(t) = \theta_z, r(t) + \theta_z, e(t)$$

$$\theta_z, r(t) \triangleq \Phi(t, 0) \theta_z, (0) + \sum_{l=1}^{t} \Phi(t, l) \Delta \hat{r}_z(l)$$

$$\theta_z, e(t) \triangleq \sum_{l=1}^{t} \Phi(t, l) \Delta \hat{e}_z(l).$$

Then, Proposition V.9 characterizes the predictive mean.

**Proposition V.9 (Prediction decomposition):** Suppose Assumptions II.1–II.3 and IV.1 hold. If $lim_{t \to \infty} \theta_d(t) = 0$ for all $j \in \mathcal{V}$, then for all $z_s \in \mathcal{Z}_{agg}$, $\mu_{\theta_z}(\theta_z, (t)) = (\sigma(\alpha_0^2)^2, \theta_z, e(t))$, where $lim_{t \to \infty} \theta_z, r(t) = \frac{1}{n} \sum_{j=1}^{n} \hat{r}_z(j)$, $\theta_z, r(t)$ is a Gaussian random variable with zero mean and $lim_{t \to \infty} \sum_{j=1}^{n} \hat{\theta}_z, t) = \frac{1}{n} \sum_{j=1}^{n} (\sigma(\alpha_0^2)^2 - \hat{r}_z)^2$.

Proof: By (14) and Line 10 of distributed GPR, we have

$$\hat{\mu}_{\theta_z}(\theta_z, (t)) = (\sigma(\alpha_0^2)^2, \theta_z, e(t)).$$

First, we show that $lim_{t \to \infty} \theta_z, r(t) = \frac{1}{n} \sum_{j=1}^{n} \hat{r}_z$. Analogous to $\theta_z(t)$, $\theta_z, r(t)$ is the solution for tracking the average of the signal $\hat{r}_z(t)$ using FODAC algorithm [32]. Since

$$\hat{r}_z = \left( \frac{\sigma^2(\alpha_0^2)^2}{\sigma^2 + (\alpha_0^2)^2} \right) \hat{r}_z, \forall t \in \mathcal{V},$$

we have $lim_{t \to \infty} \Delta \hat{r}_z(t) = 0$.

Combining this with [32, Corollary 3.1] gives

$$\lim_{t \to \infty} \theta_z, r(t) = \lim_{t \to \infty} \left( \frac{1}{n} \sum_{j=1}^{n} \hat{r}_z(j) \right) = \left( \frac{1}{n} \sum_{j=1}^{n} \hat{\theta}_z, t) \right) = \left( \frac{1}{n} \sum_{j=1}^{n} \hat{\theta}_z, t), \right)$$

Second, we show that $\theta_z, e(t)$ is a Gaussian random variable with zero mean. Note that $\theta_z, e(t) = [\theta_z, e, e(t)]^T$. Similar to $\theta_z, r(t)$, $\theta_z, e(t)$ is the solution for tracking the average of $\hat{\theta}_z, t) using FODAC$.

$$\theta_z, e(t) = \sum_{j=1}^{n} a_{ij}(t-1) \theta_z, e, e(t-1) + \Delta \hat{\theta}_z, t)$$

with initial state $\theta_z, e, e(0) = 0$. Note that $\Delta \hat{\theta}_z, t) = \sigma(\alpha_0^2)^2 \hat{\theta}_z, t) - \sigma(\alpha_0^2)^2 \hat{\theta}_z, t) (t-1)$. Recall that $\hat{\theta}_z, t), \hat{\theta}_z, t)$ are both zero-mean Gaussian random variables. Therefore, it follows from (15) that $\theta_z, e(t)$ is a Gaussian random variable with zero mean for all $t \geq 1$.

Finally, we show that $lim_{t \to \infty} \sum_{j=1}^{n} \theta_j, e(t) = \sum_{j=1}^{n} (\sigma(\alpha_0^2)^2) \theta_j, e(t)$. By Assumption II.2 and initial state $\theta_j, e(0) = \theta_j, e(0) = \theta_j, e(0) = 0$ for each $j \in \mathcal{V}$.

(15) renders

$$\lim_{t \to \infty} \sum_{j=1}^{n} \theta_j, e(t) = \lim_{t \to \infty} \sum_{j=1}^{n} \theta_j, e(t) = \lim_{t \to \infty} \sum_{j=1}^{n} \theta_j, e(t).$$

for all $t \geq 1$. Therefore, $\lim_{t \to \infty} \sum_{j=1}^{n} \theta_j, e(t) = \lim_{t \to \infty} \sum_{j=1}^{n} \theta_j, e(t)$. Combining this with (13) and the definition of $\hat{\theta}_z, t$ in Remark V.8 gives $lim_{t \to \infty} \sum_{j=1}^{n} \theta_j, e(t) = \sum_{j=1}^{n} (\sigma(\alpha_0^2)^2) \theta_j, e(t)$.

**3) Mean Analysis of Fused GPR:** This section provides the analysis of predictive mean returned by fused GPR. Recall that Lemma V.6 shows that $lim_{t \to \infty} \hat{Z}_{agg}^t(t) \neq \emptyset$. Hence, the main results in this section are Proposition V.13, where the case $lim_{t \to \infty} \hat{Z}_{agg}^t(t) \neq \emptyset$ is discussed, and Lemma V.14, where a sufficient condition for $lim_{t \to \infty} \hat{Z}_{agg}^t(t) \neq \emptyset$ is presented. Then, we discuss the case of $lim_{t \to \infty} \hat{Z}_{agg}^t(t) = \emptyset$ to conclude the proof of Theorem V.8. We first discuss the case of $lim_{t \to \infty} \hat{Z}_{agg}^t(t) \neq \emptyset$. Recall Remarks V.8 and Proposition V.9, respectively, render

$$\hat{r}_z, (t) = \hat{r}_z(t) + \hat{e}_z(t)$$

$$\hat{r}_z, (t) = \hat{r}_z(t) + \hat{e}_z(t)$$

where $\hat{r}_z(t) \triangleq (\sigma(\alpha_0^2)^2 \theta_z, r(t)$ and $\hat{e}_z(t) \triangleq (\sigma(\alpha_0^2)^2 \theta_z, e(t)$. $\theta_z, r(t)$ is zero-mean $\forall z_s \in \mathcal{Z}$. Lemma V.10 summarizes the limiting behaviors of the above variables.

**Lemma V.10:** Suppose the same conditions for Proposition V.9 hold and $d(t) \to 0 \forall j \in \mathcal{V}$. It holds that $\forall z_s \in \mathcal{Z}$

$$\hat{r}_z, (t) = \psi(t) \eta(z_s), \lim_{t \to \infty} \hat{r}_z(t) = cn(z_s)$$

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
\[ e_{z_i}^t = \psi_i^t |e_{z_i}^t|, \lim_{t \to \infty} \frac{1}{n} \sum_{j=1}^{n} e_{z_i}^j (t) = \sum_{j=1}^{n} (\mu_i^j \chi_{\mathcal{B}}^j e_{z_i}^j (t)). \]

**Proof:** Due to space limitation, the proof is included in the complete version [25].

Next we introduce necessary notations to continue the analysis. Since \( \lim_{t \to \infty} Z_{agg} (t) \neq 0 \) and Corollary V.2 hold, \( z_{agg} (t) \in \lim_{t \to \infty} \text{proj} (z_{agg}, \mathcal{Z}) (t) \) and \( g(z_{agg}) \equiv \lim_{t \to \infty} g(z_{agg}, t) \) exist. Line 11 of fused GPR gives

\[
(\mu_{i}^{[i]} (D_{i}(t)) - \eta(z_{agg}))^2 = (\mu_{i}^{[i]} (D_{i}(t)))^2 + s_{i}^{[i]} (t)
\]

where \( s_{i}^{[i]} (t) \equiv s_{z_i}^{[i]} (t) + s_{z_{agg}}^{[i]} (t) \) with

\[
s_{z_{agg}}^{[i]} (t) = 2(\mu_{i}^{[i]} (D_{i}(t)) - \eta(z_{agg}))(\mu_{i}^{[i]} (D_{i}(t)))^2 D_{i}(t)
\]

\[
s_{z_i}^{[i]} (t) = \left( e_{z_i}^t (t) \right)^2.
\]

Let \( v_{i}^{[i]} \equiv \lim_{t \to \infty} v_{i}^{[i]} (t) \), whose existence, according to its definition, is guaranteed by the existences of \( z_{agg}, g(z_{agg}) \), and Corollary V.2. Denote

\[
q_{z_{agg}}^{[i]} \equiv (\psi_{i}^{[i]} - 1)(c - \psi_{i}^{[i]}))^2 + \limsup_{t \to \infty} \mathbb{E} [e_{z_{agg}}^{[i]} (t)^2].
\]

By boundedness in Assumption IV.1 and Corollary V.2, Line 8 in fused GPR renders

\[
\lim_{t \to \infty} g_{z_{agg}}^{[i]} (t) = (1 - \psi_{i}^{[i]})(c - \psi_{i}^{[i]})k(z_{agg}, z_{agg}) \sigma_f^2
\]

\[
= -q_{z_{agg}}^{[i]}/\sigma_f^2.
\]

Combining (21) and Claim V.13.1 with (20) gives

\[
v_{i}^{[i]} = \frac{\lim_{t \to \infty} g_{z_{agg}}^{[i]} (t)}{\lim_{t \to \infty} \sigma_f^2 (\frac{1}{t})} = \frac{-q_{z_{agg}}^{[i]}/\sigma_f^2}{\lim_{t \to \infty} \sigma_f^2 (\frac{1}{t})} > 0.
\]

**Lemma V.11:** Suppose the same conditions in Theorem IV.8 Part II hold and \( d_{i}^{[i]} (t) \to 0 \) \( \forall j \in \mathcal{V} \). If \( \lim_{t \to \infty} Z_{agg} (t) \neq 0 \) for some \( z_{agg} \), then \( \lim_{t \to \infty} \mathbb{E} [s_{i}^{[i]} (t)] < 2v_{i}^{[i]} q_{z_{agg}}^{[i]} \).

**Proof:** Due to space limitation, the proof is included in the complete version [25].

**Lemma V.12:** Suppose the same conditions in Theorem IV.8 Part II hold and \( d_{i}^{[i]} (t) \to 0 \) \( \forall j \in \mathcal{V} \). If \( \lim_{t \to \infty} Z_{agg} (t) \neq 0 \) for some \( i \in \mathcal{V} \), then \( \lim_{t \to \infty} \mathbb{E} [s_{i}^{[i]} (t)] = (v_{i}^{[i]} q_{z_{agg}}^{[i]} - s_{i}^{[i]} (t)).
\]

**Proposition V.13:** Suppose the limiting behavior of \( \mathbb{E} [s_{i}^{[i]} (t)] \) when \( \lim_{t \to \infty} Z_{agg} (t) \neq 0 \).

**Proposition V.13:** Suppose the same conditions in Theorem IV.8 Part II hold and \( d_{i}^{[i]} (t) \to 0 \) \( \forall j \in \mathcal{V} \). If \( \lim_{t \to \infty} Z_{agg} (t) \neq 0 \) for some \( z_{agg} \), then \( \lim_{t \to \infty} \mathbb{E} [s_{i}^{[i]} (t)] < 0 \).

**Proof:** Denote \( b_{i}^{[i]} \equiv -2q_{z_{agg}}^{[i]}/q_{z_{agg}}^{[i]} \). Then, Lemmas V.11 and V.12 imply

\[
\lim_{t \to \infty} \mathbb{E} [s_{i}^{[i]} (t)] \leq v_{i}^{[i]} q_{z_{agg}}^{[i]} - s_{i}^{[i]} (t).
\]

We first show that \( 0 < v_{i}^{[i]} < q_{z_{agg}}^{[i]} \). The definition of \( v_{i}^{[i]} (t) \) on Line 9 in fused GPR gives

\[
v_{i}^{[i]} = \lim_{t \to \infty} \left( g_{z_{agg}}^{[i]} (t) - \frac{1}{t} \sigma_f^2 (\frac{1}{t}) \right).
\]
Taking the inverse of (22) gives \( f_2((\sigma^2_{e_i})^2))^{-1} < \left(\frac{1}{n} \sum_{i=1}^{n} f_2\left((\sigma^2_{e_i})^2\right)\right)^{-1} \leq \frac{1}{n} \sum_{i=1}^{n} f_2\left((\sigma^2_{e_i})^2\right)^{-1} \), where the last inequality follows from Lemma V.1. This gives \( \lim_{t \to \infty} \sigma^2_{\text{agg}}[D|t] \leq \lim_{t \to \infty} \left(\frac{1}{n} \sum_{i=1}^{n} \sigma^2_{\text{agg}}[D_i|t]\right) \), where the equality follows from (4). This is equivalent to

\[
\lim_{t \to \infty} \sigma^2_{\text{agg}}[D|t] > \lim_{t \to \infty} \left(\frac{\Sigma_{z_i,D_i|t}}{\varsigma_{x_i,D_i|t}}\right)^2 = \lim_{t \to \infty} \left(\frac{\Sigma_{z_i,D_i|t}}{\varsigma_{x_i,D_i|t}}\right)^2
\]

where the equality follows from Corollary V.5. Given (23) and the inequality above, since \( \varsigma_{x_i,D_i|t} \) is arbitrary, we have \( \lim_{t \to \infty} \varsigma_{x_i,D_i|t} = \emptyset \).

We now proceed to prove Part II of Theorem IV.8.

Proof of Theorem IV.8 Part II: By Lines 3 and 4 in fused GPR, it is obvious that if \( \lim_{t \to \infty} \varsigma_{x_i,D_i|t} = \emptyset \), then \( \lim_{t \to \infty} (\mu_{Z|D_i|t} - \eta(z_i))^2 = \lim_{t \to \infty} (\mu_{Z|D_i|t} - \eta(z_i))^2 \).

Proposition V.13 presents the case of \( \lim_{t \to \infty} \varsigma_{x_i,D_i|t} \neq \emptyset \), Lemma V.14 corresponds to the sufficient condition for Proposition V.13, which is the sufficient condition for \( \lim_{t \to \infty} \varsigma_{x_i,D_i|t} \neq \emptyset \).

VI. SIMULATION

In this section, we conduct Monte Carlo simulation to evaluate the developed algorithm. For the algorithms introduced below, we use (NN) to denote the version of the algorithm related to nearest-neighbor GPR and (full) to denote the version related to full GPR. We compare LiDGPR (NN), i.e., Algorithm 1, with the following five benchmarks:

1) agent-based GPR (NN), i.e., nearest-neighbor GPR (Algorithm 2);
2) agent-based GPR (full), i.e., Algorithm 2 is replaced by (2), and hence, \( \mu_{Z_i,D_i|t} = \mu_{Z_i,D_i|t}, \sigma_{Z_i,D_i|t}^2 = \left[\Sigma_{z_i,D_i|t}\right]_{z_i, z_D} \);
3) LiDGPR (full), i.e., Algorithm 1 with Algorithm 2 replaced by agent-based GPR (full);
4) centralized nearest-neighbor GPR (cNN-GPR), the centralized counterpart of LiDGPR (NN), i.e., nearest-neighbor GPR using all the data collected by all the agents;
5) centralized full GPR, i.e., (2) using all the data collected by all the agents.

The simulations are run in Python, Linux Ubuntu 18.04 on an Intel Xeon Silver 4112 CPU, 2.60 GHz with 32 GB of RAM. Consider the scenario where four mobile robots are wandering in \( Z = [1,10] \times [1,10] \) and learning spatial signals, such as temperature or wind fields. Specifically, the robots are learning ten different signals in the form \( \eta(z) = \beta \sum_{m=1}^{10} \alpha_m \sin(w_m, z_1 + w_m, z_2) \), where \( \alpha_m \sim \mathcal{N}(0, 0.01), w_m, 1 \sim \mathcal{N}(0, 1), w_m, 2 \sim \mathcal{N}(0, 1), \) and \( \beta \) is chosen such that \( \text{SNR} = \frac{\int_0^{10} |\eta(z)|^2 \, dz}{\int_0^{10} |\eta(z)|^2 \, dz} = 2. \) A realization of \( \eta \) is shown in Fig. 2(b).

For each signal, the robots repeat the trajectories for ten times, and the observations along each trajectory are subject to a different noise, where the variances of the observation noises follow \( (\sigma^2_{e_i})^2 = \sigma^2_{e} \sim \mathcal{U}(0, 0.25) \) for all \( i \in V \). Notice that there are total 100 simulations.

The communication graph of the robots is characterized by adjacency matrix \( A(t) = \frac{1 - (-1)^t}{2} \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0 \\ 0.5 & 0.5 & 0.5 & 0 \\ 0.5 & 0.5 & 0.5 & 0 \\ 0.5 & 0.5 & 0.5 & 0 \end{bmatrix} + \frac{1 + (-1)^t}{2} \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix} \), which satisfies Assumptions II.1–II.3. As shown in Fig. 2(a), the robots have spiral trajectories generated by dynamics \( \begin{bmatrix} z_i(t) \\ z_i(\theta) \end{bmatrix} = \begin{bmatrix} z_i(t) - z_i(\theta) \\ z_i(\theta) - z_i(t) \end{bmatrix} + 0.05t \begin{bmatrix} \sin(0.5t) \\ \cos(0.5t) \end{bmatrix} \), where the initial states of the robots are \( (2.5,2.5), (2.5,7.5), (7.5,7.5), \) and \( (7.5,2.5) \), respectively. Each robot \( i \) collects training data along its trajectory, i.e., \( (z_i(t), \eta(z_i(t)) + e_i(t)) \) and \( e_i(t) \sim \mathcal{N}(0, (\sigma^2_{e_i})^2) \), where \( t \geq 1 \). The set \( Z \) of test points are uniformly separated over \( Z \), and \( |Z| = 1600 \). We use 25% of the test points, uniformly separated, for the set \( Z_{\text{agg}} \), i.e., \( |Z_{\text{agg}}| = 400 \). The kernel is \( k(z, z') = \sigma^2_i \exp(-2\|z - z'\|^2) \), where \( \sigma^2_i \) is chosen following the procedure under Remark IV.2. The resulting \( \sigma^2_i \) ranges from 1.1 to 5.8 for each experiment in the Monte Carlo simulation. The prior mean is \( \mu(z) = z \) for all \( z \in Z \).

The performances of the robots are similar, and we present the figures for robot 1 due to space limitation. Let the predictive error at \( z_i \in Z \), be the distance between predictive mean and the ground truth of \( \eta \) at \( z_i \), where the distance adopts 2-norm. For example, the predictive error at \( z_i \) of agent-based GPR (NN) is \( \mu_{Z_i,D_i|t} - \eta(z_i) \). When robots’ trajectories and \( \eta \) are those in Fig. 2, Fig. 3 shows the predictive variance and predictive error over \( Z \), of cNN-GPR. We can see that the predictive variances and errors are smaller near the trajectories of the robots.

Fig. 4 shows the predictive variances and predictive errors of agent-based GPR (NN) and LiDGPR (NN) over \( Z \), of robot 1.
Fig. 4. Agent-based GPR (NN) versus LiDGPR (NN). (a) Variances of agent-based GPR (NN). (b) Errors of agent-based GPR (NN). (c) Variances of LiDGPR (NN). (d) Errors of LiDGPR (NN).

Fig. 5. Average performance of robot 1. (a) Comparison in predictive errors and variances (The upper bound in Theorem IV.8 Part 1 is scaled by 0.023). (b) $\sigma_{z_t|d_t(t)}^2$ versus Theorem IV.3 Part II (scaled by 10). (c) $\mu_{z_t|d_t(t)} - \sigma_{z_t|d_t(t)}^2$ versus Theorem IV.8 Part II (scaled by 10).

We can see that by only communicating a portion of the testing sets, LiDGPR (NN) improves the learning performances over agent-based GPR (NN) with reduced predictive variances and errors. The red dots in Fig. 4(c) and (d) are the points of $Z_{agg}$, and the “holes” indicate that the improvements take place around the trajectories (training data) of the other robots, which corresponds to the term $\kappa(\rho_{\tilde{z}_{agg}}(t))^2$ in Part II of Theorem IV.3. In addition, the improvements reduce as the test points are moving away from $Z_{agg}$, which corresponds to the terms $\kappa(\rho_{\tilde{z}_{agg}}(t))^2$ in Part II of Theorem IV.3 and $\kappa(\rho_{\tilde{z}_{agg}}(t))^2$ in Part II of Theorem IV.8, respectively.

Fig. 5(a) compares the average predictive errors and variances of LiDGPR (NN) with the five benchmarks. The $x$-axis is the iteration number, corresponding to the size of training data. The predictive variance and error at each iteration are represented by the corresponding averages over $Z_t$.

Note that the complexities in computation and memory are, respectively, $O(nt)$ and $O(nt)$ for cNN-GPR, and $O((nt)^2)$ and $O((nt)^2)$ for centralized full GPR. Notice that the differences in predictive variances and errors between cNN-GPR and centralized full GPR are small, while the diminishing rates are comparable. This shows that cNN-GPR has small performance loss compared with the benefit in reducing the complexities in computation and memory.

Comparing the curves of LiDGPR (NN) with agent-based GPR (NN) and agent-based GPR (full), we can see that LiDGPR (NN) not only compensates the information loss of using agent-based GPR (NN) to approximate agent-based GPR (full), but also gains extra information from the other robots. Note that LiDGPR (full) also shows similar improvement over agent-based GPR (full) as LiDGPR (NN) does.

Fig. 5(a) plots the theoretic error bounds in Part I of Theorems IV.3 and IV.8 over the whole Monte Carlo simulation. By multiplying by a constant, we scale down the bound by factor $0.023$ in Part I of Theorem IV.8 for better visual comparison. The orders of rates of the bounds remain the same regardless of the scaling. Comparisons between the theoretic improvement and the actual improvement of LiDGPR (NN) over agent-based GPR (NN) are shown in Fig. 5(b) and (c). Since the theoretic bounds are not tight, to make a meaningful comparison, we scale up the bounds by factor 10 in Part II of Theorems IV.3 and IV.8.

The wall clock time for prediction using LiDGPR (NN) versus $t$, the number of local data points, after linear least-square fitting, has a slope $2.13e-6$ s per test point per data point and a bias $1.19e-3$ s per test point. Recall that Section IV-F indicates that agent-based GPR (NN) has complexity $O(t)$ and agent-based GPR (full) has complexity $O(t^2)$. The growth rate of the computation times (milliseconds) of LiDGPR (NN) and LiDGPR (full) in the simulation are, respectively, $33.2t + 200$ and $0.256t^3 - 0.11t^2 - 0.512t + 27.6$. Over the simulation, the average is $1000$ test-point predictions/s, or $1$ kHz, with standard deviation $153$ predictions/s.

VII. CONCLUSION

We propose the algorithm LiDGPR that allows a group of agents to collaboratively learn a common static latent function through streaming data. The algorithm is cognizant of agents’ limited resources in communication, computation, and memory. We analyze the transient and steady-state behaviors of the algorithm and quantify the improvement brought by interagent communication. Simulations are conducted to confirm the theoretical findings. Possible future works include analysis with different implementations of agent-based GPR and the consideration of dynamic latent functions.

REFERENCES

[1] J. Levinson et al., “Towards fully autonomous driving: Systems and algorithms,” in Proc. IEEE Intell. Veh. Symp., 2011, pp. 163–168.
[2] A. Milioto, P. Lottes, and C. Stachniss, “Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs,” in Proc. IEEE Int. Conf. Robot. Autom., 2018, pp. 2229–2235.
[3] V. Vapnik, The Nature of Statistical Learning Theory. New York, NY, USA: Springer, 2013.
[4] L. G. Valiant, “A theory of the learnable,” Commun. ACM, vol. 27, no. 11, pp. 1134–1142, Nov. 1984.
[5] B. Biggio and F. Roli, “Wild patterns: Ten years after the rise of adversarial machine learning,” Pattern Recognit., vol. 84, pp. 317–331, Dec. 2018.

[6] C. K. Williams and C. E. Rasmussen, Gaussian Processes for Machine Learning, Cambridge, MA, USA: MIT Press, 2006.

[7] T. Choi and M. J. Schervish, “On posterior consistency in nonparametric regression problems,” J. Multivariate Anal., vol. 98, no. 10, pp. 1969–1987, Jan. 2007.

[8] K. Ritter, Average-Case Analysis of Numerical Problems. Berlin Heidelberg: Springer, 2007.

[9] M. P. Deisenroth, C. E. Rasmussen, and J. Peters, “Gaussian process dynamic programming,” Neurocomputing, vol. 72, no. 7–9, pp. 1508–1524, Mar. 2009.

[10] M. Mukadam, X. Yan, and B. Boots, “Gaussian process motion planning,” in Proc. Int. Conf. Robot. Autom., 2016, pp. 9–15.

[11] S. Anderson, T. D. Barfoot, C. H. Tong, and S. Särkkä, “Batch continuous-time trajectory estimation as exactly sparse Gaussian process regression,” Auton. Robots, vol. 39, no. 3, pp. 221–238, Oct. 2015.

[12] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, “When Gaussian process meets Big Data: A review of scalable GPs,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 11, pp. 4405–4423, Nov. 2020.

[13] A. V. Vecchia, “Estimation and model identification for continuous spatial processes,” J. Roy. Stat. Soc. Ser. B, vol. 50, no. 2, pp. 297–312, Jan. 1988.

[14] A. Datta, S. Banerjee, A. O. Finley, and A. E. Gelfand, “Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets,” J. Amer. Stat. Assoc., vol. 111, no. 514, pp. 800–812, Aug. 2016.

[15] G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” Neural Comput., vol. 14, no. 8, pp. 1771–1800, Aug. 2002.

[16] Y. Tresp, “A Bayesian committee machine,” Neural Comput., vol. 12, no. 11, pp. 2719–2741, Nov. 2000.

[17] J. Chen, K. H. Low, Y. Yao, and P. Jaillet, “Gaussian process decentralized control,” Automatica, vol. 46, no. 2, pp. 322–329, Feb. 2010.

[18] A. Datta, S. Banerjee, A. O. Finley, and A. E. Gelfand, “Hierarchical spatial process models with applications in geostatistics,” J. Comput. Graphical Stat., vol. 28, no. 2, pp. 401–414, Apr. 2019.

[19] T. Choi and M. J. Schervish, “On posterior consistency in nonparametric regression problems,” J. Multivariate Anal., vol. 98, no. 10, pp. 1969–1987, Jan. 2007.

[20] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, “When Gaussian process meets Big Data: A review of scalable GPs,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 11, pp. 4405–4423, Nov. 2020.

[21] A. V. Vecchia, “Estimation and model identification for continuous spatial processes,” J. Roy. Stat. Soc. Ser. B, vol. 50, no. 2, pp. 297–312, Jan. 1988.

[22] A. Datta, S. Banerjee, A. O. Finley, and A. E. Gelfand, “Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets,” J. Amer. Stat. Assoc., vol. 111, no. 514, pp. 800–812, Aug. 2016.

[23] G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” Neural Comput., vol. 14, no. 8, pp. 1771–1800, Aug. 2002.

[24] Y. Tresp, “A Bayesian committee machine,” Neural Comput., vol. 12, no. 11, pp. 2719–2741, Nov. 2000.

[25] B. Biggio and F. Roli, “Wild patterns: Ten years after the rise of adversarial machine learning,” Pattern Recognit., vol. 84, pp. 317–331, Dec. 2018.

[26] C. K. Williams and C. E. Rasmussen, Gaussian Processes for Machine Learning, Cambridge, MA, USA: MIT Press, 2006.

[27] T. Choi and M. J. Schervish, “On posterior consistency in nonparametric regression problems,” J. Multivariate Anal., vol. 98, no. 10, pp. 1969–1987, Jan. 2007.

[28] K. Ritter, Average-Case Analysis of Numerical Problems. Berlin Heidelberg: Springer, 2007.

[29] M. P. Deisenroth, C. E. Rasmussen, and J. Peters, “Gaussian process dynamic programming,” Neurocomputing, vol. 72, no. 7–9, pp. 1508–1524, Mar. 2009.

[30] M. Mukadam, X. Yan, and B. Boots, “Gaussian process motion planning,” in Proc. Int. Conf. Robot. Autom., 2016, pp. 9–15.

[31] S. Anderson, T. D. Barfoot, C. H. Tong, and S. Särkkä, “Batch continuous-time trajectory estimation as exactly sparse Gaussian process regression,” Auton. Robots, vol. 39, no. 3, pp. 221–238, Oct. 2015.

[32] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, “When Gaussian process meets Big Data: A review of scalable GPs,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 11, pp. 4405–4423, Nov. 2020.

[33] A. V. Vecchia, “Estimation and model identification for continuous spatial processes,” J. Roy. Stat. Soc. Ser. B, vol. 50, no. 2, pp. 297–312, Jan. 1988.

[34] A. Datta, S. Banerjee, A. O. Finley, and A. E. Gelfand, “Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets,” J. Amer. Stat. Assoc., vol. 111, no. 514, pp. 800–812, Aug. 2016.

[35] G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” Neural Comput., vol. 14, no. 8, pp. 1771–1800, Aug. 2002.

[36] Y. Tresp, “A Bayesian committee machine,” Neural Comput., vol. 12, no. 11, pp. 2719–2741, Nov. 2000.