Spatial Wireless Channel Prediction under Location Uncertainty

L. Srikar Muppirisetty, Tommy Svensson, Senior Member, IEEE, and Henk Wymeersch, Member, IEEE

Abstract—Spatial wireless channel prediction is important for future wireless networks, and in particular for proactive resource allocation at different layers of the protocol stack. Various sources of uncertainty must be accounted for during modeling and to provide robust predictions. We investigate two frameworks, classical Gaussian processes (cGP) and uncertain Gaussian processes (uGP), and analyze the impact of location uncertainty during both learning/training and prediction/testing phases. We observe that cGP generally fails both in terms of learning the channel parameters and in predicting the channel in the presence of location uncertainties. In contrast, uGP explicitly considers the location uncertainty and is able to learn and predict the wireless channel.

Index Terms—Gaussian processes, uncertain inputs, location uncertainty, spatial predictability of wireless channels.

I. INTRODUCTION

LOCATION-based resource allocation schemes are expected to become an essential element of emerging 5G networks, as 5G devices will have the capability to accurately self-localize and predict relevant channel quality metrics (CQM) [1]–[3] based on crowd-sourced databases. The geotagged CQM (including, e.g., received signal strength, RMS delay spread, and interference levels) from users enables the construction of a dynamic database, and this allows the prediction of CQM at arbitrary locations and future times. Current standards are already moving in this direction through the so-called minimization of drive test (MDT) feature in 3GPPP Release 10 [4]. In MDT, users collect radio measurements and associated location information in order to assess network performance.

In order to predict location-dependent radio propagation channels, we rely on mathematical models, in which the physical environment, including the locations of transmitter and receiver, play an important role. The received signal power in a wireless channel is mainly affected by three major dynamics, which occur at different length scales: path-loss, shadowing, and small-scale fading [5]. Small-scale fading decorrelates within tens of centimeters (depending on the carrier frequency), making it infeasible to predict based on location information. On the other hand, shadowing is correlated over tens of meters, depending on the propagation environment (e.g., 50–100 m for outdoor [5] and 1–2 m for indoor environments [6]). Finally, path-loss, which captures the deterministic decay of power with distance, is a deterministic function of the distance to the transmitter. Thus, based on the physical properties of the wireless channel, location-dependent models for path-loss and shadowing can be developed. With the help of spatial regression tools, these channel components can be predicted at other locations and used for resource allocation [1]. However, since localization is subject to various error sources (e.g., the global positioning system gives an accuracy of around 10 m [7] in outdoor scenarios, while ultra-wide band systems can give sub-meter accuracy), there is a fundamental need to account for location uncertainties when developing spatial regression tools.

Spatial regression tools generally comprise a training/learning phase, in which the underlying parameters are estimated based on the available training database, and a testing/prediction phase, in which predictions are made at the test locations using learned parameters and the training database. Among such tools, Gaussian processes (GP) is a powerful and commonly used regression framework, since it is generally considered to be the most flexible and provides prediction uncertainty information [8]. Two important limitations of GP are its computational complexity [9]–[12] and its sensitivity to uncertain inputs [13], [14]. To alleviate the computational complexity, various sparse GP techniques have been proposed in [9]–[11], while in [12] a connection between GP and Kalman filtering was revealed. The impact of input uncertainty was studied in [13], [14], which showed that GP was adversely affected, both in training and testing, by input uncertainties. The input uncertainty in our case corresponds to location uncertainty.

No framework has yet been developed to mathematically characterize and understand the spatial predictability of wireless channels with location uncertainty. In this paper, we build on and adapt the framework from [13], [14] to CQM prediction in wireless networks. Our main contributions are as follows:

- We show that not considering location uncertainty leads to poor learning of the channel parameters and poor prediction of CQM values at other locations, especially when location uncertainties are heterogeneous;
- We relate and unify existing GP methods that account for uncertainty during both learning and prediction, by operating directly on an input set of distributions, rather than an input set of locations;
- We describe and delimit proper choices for mean functions and covariance functions in this unified framework, so as to incorporate location uncertainty in both learning.

The authors are with the Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden, e-mail: {srikar.muppirisetty,henkw,tommy.svensson}@chalmers.se.

This research was supported, in part, by the European Research Council, under Grant No. 258418 (COOPNET), and the Swedish Research Council VR under the project Grant No. 621-2009-4555 (Dynamic Multipoint Wireless Transmission).
and prediction; and

- We demonstrate the use of the proposed framework for a spatial resource allocation application.

The remainder of the paper is structured as follows. Section III presents the channel model and details the problem description for location-dependent channel prediction with location uncertainty. In Section IV, we review channel learning and prediction in the classical GP (cGP) setup with no localization errors. Section V details learning and prediction procedures using the proposed GP framework that accounts for uncertainty on training and test locations, termed uncertain GP (uGP). Finally, numerical results are given in Section VI in addition to a resource allocation example, followed by the conclusions in Section VII.

**Notation:** Vectors and matrices are written in bold (e.g., a vector \( \mathbf{k} \) and a matrix \( \mathbf{K} \)); \( \mathbf{K}^T \) denotes transpose of \( \mathbf{K} \); \( |\mathbf{K}| \) denotes determinant of \( \mathbf{K} \); \( [\mathbf{K}]_{ij} \) denotes entry \((i,j)\) of \( \mathbf{K} \); \( \mathbf{I} \) denotes identity matrix of appropriate size; \( \mathbf{1} \) and \( \mathbf{0} \) are vectors of ones and zeros, respectively, of appropriate size; \( \mathbb{E}[\cdot] \) denotes expectation; \( \text{Cov}(\cdot) \) denotes covariance operator (i.e., \( \text{Cov}[\mathbf{y}_1, \mathbf{y}_2] = \mathbb{E}[[\mathbf{y}_1]\mathbf{y}_2^T] - \mathbb{E}[\mathbf{y}_1]\mathbb{E}[\mathbf{y}_2]^T] \)); \( \mathcal{N}(\mathbf{x}; \mathbf{m},\Sigma) \) denotes a Gaussian distribution evaluated in \( \mathbf{x} \) with mean vector \( \mathbf{m} \) and covariance matrix \( \Sigma \). Important symbols used in the paper are: \( \mathbf{x}_i \in \mathbb{R}^2 \) is an exact, true location; \( \mathbf{u}_i \in \mathbb{R}^D, D > 2 \) is a vector describing the distribution of the uncertain location; \( \mathbf{z}_i = \phi(\mathbf{u}_i) \in \mathbb{R}^2 \) is a location extracted from \( \mathbf{u}_i \) through a function \( \phi(\cdot) \) (e.g., the mean).

**II. RELATED WORK**

Prior applications of channel prediction with GP include modeling of spatial correlation of shadowing in cellular [15] and ad-hoc networks [16], as well as tracking of transmit powers of primary users in a cognitive network [17]. In [15], GP was shown to model spatially correlated shadowing to predict shadowing and path-loss at any arbitrary location. A multi-hop network scenario was considered [16], and shadowing was modeled using a spatial loss field, integrated along a line between transmitter and receiver. In [17], a cognitive network setting was evaluated, in which the transmit powers of the primary users were tracked with cooperation among the secondary users. For this purpose a distributed radio channel tracking framework using Krigeed Kalman filter was developed with location information. A study on the impact of underlying channel parameters on the spatial channel prediction variance using GP was presented in [18]. A common assumption in [15]–[18] was the presence of perfect location information.

This assumption was partially removed in [19], which extends [18] to include the effect of localization errors on spatial channel prediction. It was found that channel prediction performance was degraded when location errors were present, in particular when either the shadowing standard deviation or the shadowing correlation were large. However, [19] did not tackle combined learning and prediction under location uncertainty. The only work that explicitly accounts for location uncertainty was [20], in which the Laplace approximation was used to obtain a closed-form analytical solution for the posterior predictive distribution. However, [20] did not consider learning of parameters in presence of location uncertainty.

**III. SYSTEM MODEL**

A. Channel Model

Consider a geographical region \( \mathcal{A} \subset \mathbb{R}^2 \), where a source node is located at the origin and transmits a signal with power \( P_{\text{TX}} \) to a receiver located at \( \mathbf{x}_i \in \mathcal{A} \) through a wireless propagation channel. The received radio signal is affected mainly by distance-dependent path-loss, shadowing due to obstacles in the propagation medium, and small-scale fading due to multipath effects. The received power \( P_{\text{RX}}(\mathbf{x}_i) \) can be expressed as

\[
P_{\text{RX}}(\mathbf{x}_i) = P_{\text{TX}} g_0 \|\mathbf{x}_i\|^{-\eta} \psi(\mathbf{x}_i) |h(\mathbf{x}_i)|^2,
\]

where \( g_0 \) is a constant that captures antenna and other propagation gains, \( \eta \) is the path-loss exponent, \( \psi(\mathbf{x}_i) \) is the location-dependent shadowing and \( h(\mathbf{x}_i) \) is the component from small-scale fading. We assume measurements average small-scale fading, either in time (measurements taken over a time window) or frequency (measurements represent average power over a large frequency band). Therefore, the resulting received signal power from the source node to a receiver node \( i \) can be expressed in dB scale as

\[
P_{\text{RX}}(\mathbf{x}_i) [\text{dBm}] = L_0 - 10 \log_{10}(\|\mathbf{x}_i\|) + \Psi(\mathbf{x}_i),
\]

where \( L_0 = P_{\text{TX}} [\text{dBm}] + G_0 \) with \( G_0 = 10 \log_{10}(g_0) \) and \( \Psi(\mathbf{x}_i) = 10 \log_{10}(\psi(\mathbf{x}_i)) \). A common choice for modeling shadowing in wireless systems is through a log-normal distribution, i.e., \( \Psi(\mathbf{x}_i) \sim \mathcal{N}(0, \sigma_{\Psi}^2) \), where \( \sigma_{\Psi}^2 \) is the shadowing variance. Shadowing \( \Psi(\mathbf{x}_i) \) is spatially correlated, with well-established correlation models [21], among which the Guimondson model is widely used [22]. It is assumed that nodes measure the received power with Gaussian measurement noise. Let \( y_i \) be the scalar observation of the received power at node \( i \), which is written as \( y_i = P_{\text{RX}}(\mathbf{x}_i) + n_i \), where \( n_i \) is a zero mean additive white Gaussian noise with variance \( \sigma_n^2 \). For the sake of notational simplicity, we do not consider a three-dimensional layout, the impact of non-uniform antenna gain patterns, or distance-dependent path-loss exponents.

B. Location Error Model

In practice, nodes may not have access to their true location \( \mathbf{x}_i \), but only to a distribution \( p(\mathbf{x}_i) \). These distributions can be described by a finite set of parameters, \( \mathbf{u}_i \in \mathbb{R}^D, D > 2 \), e.g., a mean and a covariance matrix for Gaussian distributions. The set of all valid distributions is denoted by \( \mathcal{U} \subset \mathbb{R}^D \). Within this set, the set of all delta Dirac distributions over locations is denoted by \( \mathcal{X} \subset \mathcal{U} \). Note that \( \mathcal{X} \) is equivalent to the set \( \mathcal{A} \) of possible locations. Finally, we introduce a function \( \phi: \mathcal{U} \rightarrow \mathcal{A} \) that extracts a position from the distribution (in our case chosen as the mean), and denote \( \mathbf{z}_i = \phi(\mathbf{u}_i) \in \mathcal{A} \).

1Vector measurements are also possible (e.g., from multiple base stations), but not considered here for the sake of clarity.
C. Problem Statement

We assume a central coordinator, which collects a set of received power measurements $y = [y_1, \ldots, y_N]^T$ with respect to a common source from $N$ nodes, along with their corresponding location distributions $U = \{u_i\}_{i=1}^N$. Our goals are to perform

1) **Learning**: construct a spatial model (through estimating model parameters $\Theta$, to be defined later) of the received power based on the measurements; 2) **Prediction**: determine the predictive distribution $p(P_{RX}(x_*)|y, U, \Theta, x_*)$ of the power at test locations $x_*$ and $p(P_{RX}(u_*)|y, U, \Theta, u_*)$ for test location distributions $u_*$. We will consider two methods for learning and prediction (shown in Fig. 1): classical GP (Section IV), which ignores location uncertainty and only considers $z_i = \phi(u_i)$, and uncertain GP (Section V), which is a method that explicitly accounts for location uncertainty.

IV. Channel Prediction with Classical GP

We first present cGP under the assumption that all locations during learning and prediction are known exactly. Later in this section, we will discuss the impact of location uncertainties on cGP in learning/training and prediction/testing.

A. cGP without Location Uncertainty

We designate $x_i \in A$ as the **input** variable, and $P_{RX}(x_i)$ as the **output** variable. We model $P_{RX}(x_i)$ as a GP with mean function $\mu(x_i) : A \rightarrow \mathbb{R}$ and a positive semidefinite covariance function $C(x_i, x_j) : A \times A \rightarrow \mathbb{R}^+$, and we write

$$P_{RX}(x_i) \sim \mathcal{GP}(\mu(x_i), C(x_i, x_j)).$$  \hfill (3)

The mean function is defined as $\mu(x_i) = \mathbb{E}_{\phi(x_i)}[P_{RX}(x_i)] = L_0 - 10 \eta \log_{10}(||x_i||)$, where the last transition follows from (2). The covariance function is defined as $C(x_i, x_j) = \text{Cov}[P_{RX}(x_i), P_{RX}(x_j)]$. We will consider a class of covariance functions of the form:

$$C(x_i, x_j) = \sigma_y^2 \exp\left(-\frac{||x_i - x_j||^p}{d_x^p}\right) + \delta_{ij} \sigma_{\text{proc}}^2,$$  \hfill (4)

where $\delta_{ij} = 1$ for $i = j$ and zero otherwise, $p \geq 1$, and $\sigma_{\text{proc}}$ captures any noise variance term that is not due to measurement noise (more on this later). Note that both the mean function and the covariance depend on parameters $\Theta = [\sigma_y, \sigma_{\text{proc}}, d_x, L_0, \eta, \sigma_y]$, \hfill (5)

that may not be known a priori.

\footnote{Here, $P_{RX}(u_*)$ should be interpreted as the expected received power, $p(P_{RX}(u_*)|y, U, \Theta, u_*) = \int p(P_{RX}(x_*)|y, U, \Theta, x_*)p(x_*)dx_*$, where $p(x_*)$ is described by $u_*$.}

\footnote{We have considered spatial interpolation only for simplicity, but extending the model to incorporate temporal interpolation is straightforward.}

Figure 1. Learning and prediction phases of cGP and uGP. The difference in learning in uGP compared to cGP is that it considers location uncertainty of the nodes. The estimated model parameters $\Theta$ are derived during the learning phase. The mean $\bar{P}_{RX}(x_*)$ and variance $P_{RX}(x_*)$ of the posterior predictive distribution in cGP corresponds to a location $x_*$ extracted from the distribution $u_*$. In contrast, the mean $P_{RX}(u_*)$ and variance $P_{RX}(u_*)$ of the posterior predictive distribution in uGP pertains to the entire location distribution $u_*$. 

1) **Learning**: The objective during learning is to infer the model parameters $\Theta$ from observations at **known** locations. We introduce $X = [x_1^T, x_2^T, \ldots, x_N^T]^T$ as the collection of $N$ measurement locations and recall that $y$ is the vector of noisy received power measurements at those locations. The resulting training database is thus $\{x_i, y_i\}$. Due to the GP model, the joint distribution of the $N$ training observations exhibits a Gaussian distribution $\mathcal{N}$,

$$p(y|X, \Theta) = \mathcal{N}(y; \mu(X), K),$$  \hfill (6)

where $\mu(X) = [\mu(x_1), \mu(x_2), \ldots, \mu(x_N)]^T$ is the mean vector and $K$ is the covariance matrix of the received powers with entries $K_{ij} = C(x_i, x_j) + \sigma_y^2 \delta_{ij}$. The model parameters can be learned through maximum likelihood estimation, given the training database, by minimizing the negative log-likelihood function with respect to $\Theta$: \hfill (7)

$$\Theta = \arg\min_{\Theta} \{-\log(p(y|X, \Theta))\}.$$ 

The negative log-likelihood function is usually not convex and contains multiple local optima that might not explain the observations properly. Additional details on the learning process are provided later. Once $\Theta$ is estimated from $\{X, y\}$, the training process is complete.

2) **Prediction**: After $\Theta$ is obtained, we can determine the predictive distribution of $P_{RX}(x_*)$ at a new and arbitrary test location $x_*$, given the training database $\{X, y\}$. We first form the joint distribution as \hfill (8)

$$\begin{bmatrix} y \\ P_{RX}(x_*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(X) \\ K_{kk} \end{bmatrix}, \begin{bmatrix} K & k_\ast \\ k_\ast^T & k_{ss} \end{bmatrix}\right),$$

where $k_s$ is the $N \times 1$ vector of cross-covariances $C(x_s, x_i)$ between the received power at $x_s$ and at the training locations $x_i$, and $k_{ss}$ is the prior variance, given by $C(x_s, x_s)$. Conditioning on the observations $y$, we obtain the Gaussian posterior predictive distribution $p(P_{RX}(x_*)|X, y, \Theta, x_*)$...
for the test location \( x_* \). The mean (\( \tilde{P}_{RX}(x_*) \)) and variance (\( \tilde{P}_{RX}(x_*)^2 \)) of this distribution turn out to be

\[
\tilde{P}_{RX}(x_*) = \mu(x_*) + k_*^{T}K^{-1}(y - \mu(X))
\]

\[
= \mu(x_*) + \sum_{i,j=1}^{N} [K^{-1}]_{ij} (y_j - \mu(x_j)) C(x_*, x_i)
\]

\[
= \mu(x_*) + \sum_{i=1}^{N} \beta_i C(x_*, x_i).
\]

\[
\tilde{P}_{RX}(x_*) = k_*^{T}K^{-1}k_*
\]

\[
= k_* - \sum_{i,j=1}^{N} [K^{-1}]_{ij} C(x_*, x_i) C(x_*, x_j),
\]

where \( \beta_i = \sum_{j=1}^{N} [K^{-1}]_{ij}(y_j - \mu(x_j)) \). In (9), \( \mu(x_*) \) corresponds to the deterministic path-loss component at \( x_* \), which is corrected by a term involving the database and the correlation between the measurements at the training locations and the test location. In (10), \( k_* \) is the prior variance (i.e., the variance in the absence of measurements), which is reduced by a term that accounts for the correlation of nearby measurements.

B. cGP with Location Uncertainty

Now let us consider the case when the nodes do not have access to their true location \( x \), but only to a distribution \( p(x) \), which is described by \( u \in U \). Fig. 2 illustrates the impact of location uncertainties assuming location errors are from a Gaussian distribution, depicting the measurements as a function of \( z = \phi(u) \) for three different types of uncertainty: high location uncertainty, medium location uncertainty, and low location uncertainty. The input uncertainty manifests itself as output noise, which is captured by \( \sigma_{\text{proc}} \) with a magnitude dependent on the location uncertainty. This noise must be accounted for in the model when learning \( \Theta \) and also when predicting the posterior distribution in a test input \( u_* \). When these uncertainties are ignored, both learning and prediction will be of poor quality, as described below.

1) Learning from uncertain training locations: In this case the training database comprises noisy locations \( \mathbf{z}_i = \phi(u_i) \) and the power measurements \( y_i \) at the true but unknown training locations \( x_i \neq u_i \) (see also Fig. 1 top). The measurements will be of the form shown in Fig. 2. The model parameters \( \Theta \) to explain all these measurements must either (i) assign very short correlation distances \( \phi \), large \( \sigma_{\phi} \), and small \( \sigma_{\text{proc}} \), as some seemingly nearby events will appear uncorrelated, or (ii) it must assign longer correlation distances, smaller \( \sigma_{\phi} \), and explain the measurements by assigning a higher value to \( \sigma_{\text{proc}} \). In the first case, the estimate \( \hat{\Theta} \) will lead to poor predictions at any (exact) test location \( x_* \), with a posterior mean close to the prior mean \( \mu(x_*) \) and a posterior variance close to the prior variance \( k_* \). In the second case, predictions will be better, as correlations can be exploited to reduce the posterior variance. However, the model must explain different levels of input uncertainty with a single covariance function, which can make no distinctions between locations with low, medium, or high uncertainty. This will lead to poor performance when location error statistics differ from node to node.

2) Prediction at an uncertain test location: In the case where training locations are exactly known (i.e., \( z_i = x_i \), \( \forall i \)), we may want to predict the power at an uncertain test location \( u_* \), made available to cGP in the form \( z_* = \phi(u_*) \), while the true test location \( x_* \) is not known and generally \( x_* \neq z_* \). This scenario can occur when a mobile user relies on a low-quality localization system and reports an erroneous location estimate to the base station (BS). The wrong location has impact on the predicted posterior distribution since the predicted mean \( \mu(z_*) \) will differ from the correct mean \( \mu(x_*) \).

In addition, \( k_* \) will contain erroneous entries: the \( j \)-th entry will be too small when \( \|z_* - x_j\| > \|x_* - x_j\| \) and too large when \( \|z_* - x_j\| < \|x_* - x_j\| \). This will affect both the posterior mean \( \tilde{\mu}(z_*) \) and variance \( \tilde{\sigma}(z_*) \).

In the case were training locations are also unknown, i.e., \( Z \neq X \), where \( Z = [z_1^T, z_2^T, \ldots, z_N^T]^T \), and \( z_* \neq x_* \), these effects are further exacerbated by the improper learning of \( \Theta \).

V. CHANNEL PREDICTION WITH UNCERTAIN GP

In the previous section, we have argued that cGP is unable to learn and predict properly when training or test locations are not known exactly, especially when location error statistics are heterogeneous. In this section, we explore several possibilities to explicitly incorporate location uncertainty. We recall that \( U \) denotes the set of all distributions over the locations in the environment \( \mathcal{A} \), while \( \mathcal{X} \subset U \) represents the delta Dirac distributions over the positions, and has a one-to-one mapping to \( \mathcal{A} \).

We will first describe a Bayesian approach where the uncertain input is marginalized, leading to a non-Gaussian output distribution. We then derive a Gaussian approximation of the output distribution through moment matching and detail the corresponding learning and prediction expressions. From these expressions, the concepts of expected mean function
and expected covariance function naturally appear. Third, we discuss uncertain GP, which is a Gaussian process with input \( u \) from input set \( U \) and output \( y \). Finally, we relate these three approaches in a unified view. For each method, we detail the quality of the solution and the computational complexity. We note that other approaches exist, e.g., through linearizing the output around the mean of the input \([23], [24]\), but they are limited to mildly non-linear scenarios.

A. Bayesian Approach

In a Bayesian context, we learn and predict by integrating the respective distributions over the uncertainty of the training and test locations. As this method will involve Monte Carlo integration, we will refer to it as Monte Carlo GP (MCGP).

1) Learning: The likelihood function with uncertain training and test locations \( p(y|U, \Theta) \) is obtained by integrating \( p(y|X, \Theta) \) over the random training locations:

\[
p(y|U, \Theta) = \int p(y|X, \Theta) p(X) \, dX, \tag{11}
\]

where \( p(X) = \prod_{i=1}^{N} p(x_i) \). There is generally no closed-form expression for the integral \([11]\). The integral can be approximated using a Monte Carlo approach by drawing \( M \) samples from \( X^{(m)} \sim p(X) \) as

\[
p(y|U, \Theta) \approx \frac{1}{M} \sum_{m=1}^{M} p(y|X^{(m)}, \Theta) = \frac{1}{M} \sum_{m=1}^{M} \mathcal{N}(y; \mu(X^{(m)}), K^{(m)}), \tag{12}
\]

where \( [K^{(m)}]_{ij} = C(x_i^{(m)}, x_j^{(m)}) + \sigma_\text{u}^2 \delta_{ij} \) and \( \mu(X^{(m)}) = [\mu(x_1^{(m)}), \mu(x_2^{(m)}), \ldots, \mu(x_N^{(m)})]^T \). Finally, an estimate of \( \Theta \) can be found by minimizing the negative log-likelihood function

\[
\tilde{\Theta} = \arg\min_{\Theta} \left\{ -\log(p(y|U, \Theta)) \right\}, \tag{13}
\]

which has to be solved numerically.

Remark 1. This optimization involves high computational complexity and possibly numerical instability (due to the sum of exponentials). More importantly, a good estimate of \( \Theta \) can only be found if a sample \( X^{(m)} \) is generated that is close to the true locations \( X \). As this is very unlikely even for large \( M \), we expect \([13] \) to lead to poor estimates.

2) Prediction: The posterior predictive distribution with uncertain training and test locations \( p(P_{RX}(u_i)|U, y, \tilde{\Theta}, u_i) \) is obtained by integrating \( p(P_{RX}(x_*)|X, y, \tilde{\Theta}, x_*) \) with the distribution of the uncertain training and test locations

\[
p(P_{RX}(u_i)|U, y, \tilde{\Theta}, u_i) = \int p(P_{RX}(x_*)|X, y, \tilde{\Theta}, x_*) p(X) p(x_*) \, dX \, dx_*, \tag{14}
\]

The integral is again analytically intractable in general. For analytical expression of the integral, \([20]\) makes use of the Laplace approximation. The integral can also be approximated by drawing \( M \) samples of \( X^{(m)} \sim p(X) \) and \( x_*^{(m)} \sim p(x_*) \) as

\[
p(P_{RX}(u_i)|U, y, \tilde{\Theta}, u_i) \approx \frac{1}{M} \sum_{m=1}^{M} p(P_{RX}(x_*^{(m)})|X^{(m)}, y, \tilde{\Theta}, x_*^{(m)}) = \frac{1}{M} \sum_{m=1}^{M} \mathcal{N}(P_{RX}(x_*^{(m)}); \tilde{P}_{RX}(x_*^{(m)}), \tilde{P}_{RX}(x_*^{(m)})) \tag{15}
\]

As the number of samples \( M \) increases, the approximate distribution will tend to the true distribution. We refer to \([13] - [15] \) as Monte Carlo GP (MCGP). From \([15] \), we can compute the mean \( (\tilde{P}_{RX}^{MC}(u_*)) \) and the variance \( (\tilde{P}_{RX}^{MC}(u_*)) \) as

\[
\tilde{P}_{RX}^{MC}(x_*) = \frac{1}{M} \sum_{m=1}^{M} \tilde{P}_{RX}(x_*^{(m)}) \tag{16}
\]

\[
\tilde{P}_{RX}^{MC}(u_*) = \frac{1}{M} \sum_{m=1}^{M} \left( \tilde{P}_{RX}(x_*^{(m)}) - \tilde{P}_{RX}(u_*) \right)^2 + \frac{1}{M} \sum_{m=1}^{M} \tilde{P}_{RX}(x_*^{(m)}). \tag{17}
\]

Remark 2. Prediction is numerically straightforward, though it involves the inversion of an \( N \times N \) matrix \( K \) for each training sample \( X^{(m)} \). In the case training locations are known, we can utilize cGP to obtain a good estimate of \( \Theta \) and efficiently and accurately compute \( P_{RX}^{MC}(u_*) \) and \( \tilde{P}_{RX}^{MC}(u_*) \). When both training and test locations are known, the above procedure reverts to cGP.

B. Gaussian Approximation

While MCGP can deal with location uncertainty during prediction, it will fail to deliver adequate estimates of \( \Theta \) during learning. To address this, we can modify \( p(y|U, \Theta) \) from \([11]\) using a Gaussian approximation through moment matching. In addition, we can also form a Gaussian approximation of \( p(P_{RX}(u_i)|U, y, \tilde{\Theta}, u_i) \) for prediction. We will term this approach Gaussian approximation GP (GAGP). The expressions that are obtained in the learning of GAGP, namely the expectation of mean and covariance functions will be used later in the design of uncertain GP (described in Section V.C).

1) Learning: The mean of \( p(y|U, \Theta) \) is given by

\[
\mathbb{E}[y|U, \Theta] = \int y \, p(y|X, \Theta) \, p(X) \, dX \, dy = \int \mu(X) \, p(X) \, dX = \mu(U), \tag{18}
\]

where \( \mu(U) = [\mu(u_1), \mu(u_2), \ldots, \mu(u_N)]^T \) and \( \mu(u_i) = \int \mu(x_i) \, p(x_i) \, dx_i \). The covariance matrix of \( p(y|U, \Theta) \) can
be expressed as
\[
\text{Cov}[y, y|U, \Theta] = \int y y^T p(y|U, \Theta) p(U) dU - \mu(U) \mu(U)^T
\]
\[
= \left( (K + \mu(X) \mu(X)^T) p(X) dX - \mu(U) \mu(U)^T \right)
\]
\[
= K_u + \Delta,
\] (19)
where \([K_u]_{ij} = C_u(u_i, u_j) + \sigma_n^2 \delta_{ij}\) in which
\[
C_u(u_i, u_j) = \int C(x_i, x_j) p(x_i) p(x_j) dx_i dx_j
\]
and \(\Delta\) is a diagonal matrix with entries
\[
[\Delta]_{ii} = \int \mu^2(x_i) p(x_i) dx_i - \mu^2(u_i).
\] (21)
These approximations allow us to express the likelihood function as \(p(y|U, \Theta) \approx\mathcal{N}(y; \mu(U), K_u + \Delta)\), so that \(\Theta\) can be estimated by minimizing the negative log-likelihood function
\[
\hat{\Theta} = \arg\min_{\Theta} \left\{ -\log(\mathcal{N}(y; \mu(U), K_u + \Delta)) \right\}.
\] (22)

Remark 3. Learning in GAGP involves computation of the expected mean in (18) and (21), as well as the expected covariance function in (20). These integrals are generally again intractable, but there are cases where closed-form expression exist [13, 14]. These will be discussed in detail in Section 3.4. GAGP avoids the numerical problems present in MCGP and will generally be able to provide a good estimate of \(\Theta\).

2) Prediction: We will approximate the predictive distribution \(p(\hat{y}|u, y, \Theta)\) by a Gaussian with mean \(\tilde{P}_{\text{RX}}^{GA}(u_*)\) and variance \(\tilde{P}_{\text{RX}}^{GA}(u_*)\). These are given by
\[
\tilde{P}_{\text{RX}}^{GA}(u_*) = \mu(u_*) + \sum_{i=1}^N \beta_i \int C(x_*, x_i) p(x_i) dx_i
\] (23)
\[
\text{and} \quad \tilde{P}_{\text{RX}}^{GA}(u_*)^2 = k_* - \sum_{i,j=1}^N [K^{-1}]_{ij} \int C(x_*, x_i) C(x_*, x_j) p(x_i) dx_i
\]
\[
+ \int \mu(x_*)^2 p(x_*) dx_* + 2 \sum_{i=1}^N \beta_i \left( \int \mu(x_*) C(x_*, x_i) \right)
\]
\[
\times p(x_*) dx_* + \sum_{i,j=1}^N \beta_i \beta_j \int C(x_*, x_i) C(x_*, x_j) p(x_i) dx_i
\]
\[
- \tilde{P}_{\text{RX}}^{GA}(u_*)^2.
\] (26)

Remark 5. In case training locations are known \(U \subset \mathcal{X}\), so that (23) reverts to
\[
\tilde{P}_{\text{RX}}^{GA}(u_*) = \mu(u_*) + \sum_{i=1}^N \beta_i \int C(x_*, x_i) p(x_i) dx_i
\] (25)
and (24) becomes
\[
= k_* - \sum_{i=1}^N [K^{-1}]_{ij} \int C(x_*, x_i) C(x_*, x_j) p(x_i) dx_i
\]
\[
+ \int \mu(x_*)^2 p(x_*) dx_* + 2 \sum_{i=1}^N \beta_i \left( \int \mu(x_*) C(x_*, x_i) \right)
\]
\[
\times p(x_*) dx_* + \sum_{i,j=1}^N \beta_i \beta_j \int C(x_*, x_i) C(x_*, x_j) p(x_i) dx_i
\]
\[
- \tilde{P}_{\text{RX}}^{GA}(u_*)^2.
\]
both of which can be computed in closed form for certain cases in which the function \(\mu(x)\) is constant [14 Section 3.4]. When both \(U \subset \mathcal{X}\) and \(u_* \subset \mathcal{X}\), GAGP reverts to cGP.

C. Uncertain GP

While GAGP avoids the problems during learning inherent to MCGP, prediction is generally intractable. Hence, GAGP is not a fully coherent approach to deal with location uncertainty. To address this, we consider a new type of GP (uGP), which operates directly on the location distributions, rather than on the unknown locations. uGP involves a mean function \(\mu_{uGP}(u_i) : U \to \mathbb{R}\) and a positive semidefinite covariance function \(C_{uGP}(u_i, u_j) : U \times U \to \mathbb{R}^+\), which considers as input set \(U\) and output \(y\). In other words,
\[
P(\hat{y}|u_i) \sim \mathcal{G}P(\mu_{uGP}(u_i), C_{uGP}(u_i, u_j)).
\] (27)
The mean function is given by \(\mu_{uGP}(u_i) = \mathbb{E}_y[\mathbb{E}_{\Phi(x_i)}[P(\hat{y}|x_i)]]\), already introduced as the expected mean function in (13). However, for the mean function to be useful in a GP context, it should be available in closed form. As in cGP, we have significant freedom in our choice of covariance function. Given the observations from Fig. 2 it is desirable to have a covariance function that (i) is available in closed form; (ii) leads to decreasing correlation with increasing input uncertainty (even when both inputs have same mean); (iii) can account for varying amounts of input uncertainty; (iv) reverts to a covariance function of the form (4) when \(u \subset \mathcal{X}\); (v) does not depend on the mean function \(\mu(x)\).

The mean function: According to law of iterated expectations, the mean function \(\mu(u_i)\) is expressed as
\[
\mu(u_i) = L_0 - \eta \mathbb{E}_{\mathcal{X}_i} [\log_{10} (||x_i||)].
\] (28)
While there is no closed-form expression available for (28) we can form a polynomial approximation \(\sum_{j=0}^J a_j ||x_i||^j \approx \log_{10} (||x_i||)\), solved by least squares minimization. For a given range of \(||x_i||\), this approximation can be made arbitrarily close by increasing the order \(J\). When \(p(||x_i||)\) is approximately
Gaussian (which may be the case for \( \|x_i\| \gg 0 \)), \( \mu(u_i) \approx -L_0 - 10 \eta \sum_{j=0}^{J} a_j \mathbb{E}_x[\|x_i\|^2] \) can be evaluated exactly, since all Gaussian moments are known. See Appendix \( \text{A} \) for details on the approximation.

The covariance function: While any covariance function meeting the criteria (i)–(v) listed above can provide good predictions, a natural choice is the expected output covariance, i.e., choose \( C_{\alpha x}(u_i, u_j) = \text{Cov}[\hat{P}_{RX}(x_i), \hat{P}_{RX}(x_j)] \) for GAGP (22) in GAGP for learning (22). \( \text{Cov} \) is the variance of the posterior predictive distribution \( \hat{P}_{RX}(x_i) \). Unfortunately, as we can see from (19), this choice does not satisfy criterion (v). An alternative choice is the expectation of \( C_{\alpha x}(u_i, u_j) \) given by (20). This choice clearly satisfies criteria (ii), (iii), (iv), and (v). To satisfy (i), we can select appropriate covariance functions, tailored to the distributions \( p(x_i) \), or appropriate distributions \( p(x_i) \) for a given covariance function. Examples include:

- Polynomial covariance functions for Gaussian \( p(x_i) \) \[13], \[14].
- Covariance functions of the form \[4] with \( p = 1 \), \( x_i \in \mathbb{R} \), for exponential covariance.
- Covariance functions of the form \[4] with \( p = 2 \), \( x_i \in \mathbb{R}^2 \), for Gaussian \( p(x_i) \) (i.e., \( p(x_i) = N(x_i; z_i, \Sigma_i) \)). The expected covariance function is then given by \[13], \[14].

\[
C_{\alpha x}^{SE}(u_i, u_j) = \delta_{ij} \sigma^2_{\text{proc}} + \sigma^2_\phi \left| I + d_c^{-2}(\Sigma_i + \Sigma_j)(1 - \delta_{ij}) \right|^{-1/2} \times \exp \left( -(z_i - z_j)^T (d_c^2 I + \Sigma_i + \Sigma_j)^{-1}(z_i - z_j) \right).
\]

Note that the factor \( \left| I + d_c^{-2}(\Sigma_i + \Sigma_j)(1 - \delta_{ij}) \right|^{-1/2} \) ensures that inputs \( i \neq j \) with the same mean exhibit lower correlation with increasing uncertainty. The factor \( (d_c^2 I + \Sigma_i + \Sigma_j)^{-1} \) ensures that the good measurements can be explained by a reasonable value of \( d_c \), while for poor measurements (with high input uncertainty) any reduction in correlation distance is absorbed through the covariances.

1) Learning: With the choices for \( \mu_{\text{GP}}(u_i) = \mu(u_i) \) and \( C_{\text{GP}}(u_i, u_j) = C_{\alpha x}(u_i, u_j) \), the model parameters are found by minimizing the log-likelihood function

\[
\hat{\Theta} = \arg \min_{\Theta} \{ -\log(p(y|U, \Theta)) \}
= \arg \min_{\Theta} \{ -\log(N(y; \mu(U), K_u)) \}.
\]

Note that in contrast to GAGP, we have constructed uGP so that \( \mu(U) \) and \( K_u \) are available in closed form, making numerical minimization tractable.

Remark 6. Learning of uGP (30) corresponds to the case of learning (22) in GAGP for \( \Delta = 0 \) (e.g., for constant mean processes).

2) Prediction: Let \( \tilde{P}_{RX}(u_i) \) be the mean and \( \tilde{P}_{RX}(u_i) \) be the variance of the posterior predictive distribution \( p(\hat{P}_{RX}(u_i)|U, y, \Theta, u_i) \) of uGP with uncertain training and test locations, then \( p(\hat{P}_{RX}(u_i)|U, y, \Theta, u_i) = \mathcal{N}(\tilde{P}_{RX}(u_i); \tilde{P}_{RX}(u_i), \tilde{P}_{RX}(u_i)) \). The expressions for \( \tilde{P}_{RX}(u_i) \) and \( \tilde{P}_{RX}(u_i) \) are now in standard GP form:

\[
\tilde{P}_{RX}(u_i) = \mu(u_i) + k_{u_i}^T K_u^{-1}(y - \mu(U)) \quad (31)
\]

\[
\tilde{P}_{RX}(u_i) = k_{u_i} - k_{u_i}^T K_u^{-1} k_{u_i} \quad (32)
\]

where \( k_{u_i} \) is the \( N \times 1 \) vector of cross-covariances \( C_{\alpha x}(u_i, u_i) \) between the received power at the test distribution \( u_i \) and at the training distribution \( u_i \), and \( k_{u_i} \) is the a priori variance \( C_{\alpha x}(u_i, u_i) \).

Remark 7. In case \( U \subset X \), the mean \( \tilde{P}_{RX}(u_i) \) and the variance \( \tilde{P}_{RX}(u_i) \) can be obtained from the expressions (31) and (32) respectively by letting \( \Sigma_i = 0, \forall i \in \{1, 2, \ldots, N\} \). Furthermore, the resulting mean \( \tilde{P}_{RX}(u_i) \) is exactly the same as (25), obtained in GAGP. However, due to a different choice of covariance function, the predicted variance \( \tilde{P}_{RX}(u_i) \) is different from (26).

Remark 8. When \( u_i \in X \), the mean \( \tilde{P}_{RX}(u_i) \) and the variance \( \tilde{P}_{RX}(u_i) \) are obtained from (31) and (32) by setting \( \Sigma_i = 0 \).

D. Unified View

A unified view of the four methods (cGP, MCGP, GAGP, and uGP) is provided in Fig. 3 all four methods revert to cGP when training and predictions occur in \( X \). MCGP is able to consider general input distributions in \( U \), but leads to non-Gaussian output distributions. Through a Gaussian approximation of these output distributions, GAGP can consider general inputs and directly determine a Gaussian output distribution. Both of these approaches (MCGP and GAGP) have in common that they treat the process with input \( x \in A \) as a GP. In contrast, uGP treats the process with input \( u \in U \) as a GP. This allows for a direct mapping from inputs in \( U \) to Gaussian output distributions. In terms of tractability for learning and prediction, the four methods are compared in Table 1. We see that among all four methods, uGP combines tractability with good performance.

![Figure 3. Relation between cGP, MCGP, GAGP, and uGP. All methods are equivalent when the input is limited to \( X \) (grey shaded area).](image_url)
VI. NUMERICAL RESULTS AND DISCUSSION

In this section, we show learning and prediction results of cGP, uGP, and MCGP with uncertainty in training and/or test locations. In Section VI.D, we describe a resource allocation problem, where communication rates are predicted at future locations using cGP and uGP, in the presence of location uncertainty during training.

Table I

| Method   | Learning          | Prediction      |
|----------|-------------------|-----------------|
| cGP      | tractable, poor quality | closed-form, poor quality |
| MCGP     | complex, poor quality | tractable       |
| GAGP     | tractable in some cases | intractable     |
| uGP      | tractable by design  | closed-form     |

Figure 4. Impact of location uncertainty on learning the hyperparameters using cGP, uGP, and MCGP. The hyperparameters are estimated for each value of the mean location error standard deviation and for 40 realizations of the channel field. Results shown are the mean estimate of the hyperparameters and error bars with one standard deviation. Impact of location uncertainty in shown when estimating: (a) $d_c$, (b) $\sigma_\Psi$, (c) $\sigma_{proc}$, (d) $\eta$.

Table II

| Parameter | Value | Parameter | Value |
|-----------|-------|-----------|-------|
| $\eta$    | 2.5   | $M$       | 300   |
| $\sigma_0$| 0.01  | $W$       | -100 dBm |
| $d_c$     | 15 m  | $\sigma_\Psi$ | 10 dB |

A. Simulation Setup

A geographical region $\mathcal{A}$ is considered and a base station is placed at the origin. A one dimensional radio propagation field is generated with sampling locations at a resolution of 0.25 m using an exponential covariance function $C_{ref}(x_i, x_j) = \sigma_\Psi^2 \exp\left(-||x_i - x_j||/d_c\right)$. The simulation parameters used to obtain the numerical results are given in Table II.

We assume isotropic localization errors, so that $\Sigma_i = \sigma_i^2 I$. To capture the effect of heterogeneous location errors, we draw the location error standard deviations from an exponential distribution, i.e., $\sigma_i \sim \text{Exp}(\lambda)$, where $\lambda$ is the average
Appendix B.

and MCGP. The learning of the hyperparameters is detailed in mismatch in the covariance function, we absorb this mismatch in \( \sigma_{\text{proc}} \), which is learned offline (more on this in Appendix B). We assume nodes know \( \sigma_n \) and \( L_0 \) from their measurements, therefore they are not estimated as part of the learning process.

B. Learning

---

Fig. 4 depicts the impact of location uncertainty on the learning of hyperparameters \([d_c, \sigma\Phi, \sigma_{\text{proc}}, \eta]\) for cGP, uGP, and MCGP. The learning of the hyperparameters is detailed in Appendix B.

- **cGP:** We first consider a variant of cGP, denoted as cGP-no-proc, in which \( \sigma_{\text{proc}} \) is fixed to zero. In cGP-no-proc, when \( \lambda = 0 \), the estimate \( \hat{d}_c \) is non-zero. However, it can be observed in Fig. 4(a), that with increase in \( \lambda \), \( \hat{d}_c \) decreases quickly to zero. Hence, cGP-no-proc will model the GP as a white process with high variance \( \sigma_q^2 \) and thus cannot handle the location uncertainty. On the other hand, in cGP where we estimate \( \sigma_{\text{proc}} \), \( \hat{\sigma}_{\text{proc}} \) absorbs part of location uncertainty (see Fig. 4(c)). Consequently, the part of the observations that must be explained through \( \sigma_q \) is reduced, leading to a reduction of \( \hat{\sigma}_q \) with \( \lambda \). Due to this, cGP considers the measurements constitute a slowly varying process, therefore \( \hat{d}_c \) increases with \( \lambda \). An interesting observation is that the error bars for \( \hat{d}_c \) also increase with \( \lambda \). Hence, among cGP-no-proc and cGP, only cGP can reasonably deal with location uncertainty.

- **MCGP:** The behavior is similar to that of cGP, i.e., an increase in \( \hat{d}_c \) and a decrease in \( \hat{\sigma}_q \), when increasing in \( \lambda \). However, \( \hat{\sigma}_q \) decreases more quickly with \( \lambda \) when compared to cGP. These effects can be attributed to two causes: first of all, the inherent problem of drawing a finite number of samples as detailed at the end of Section V.A1; secondly, the fluctuations in the estimated path loss exponent \( \hat{\eta} \) with increasing \( \lambda \) (see Fig. 4(d)). The error bars of the estimates in this case are even higher than in cGP.

- **uGP:** As mentioned before, in uGP \( \sigma_{\text{proc}} \) is determined offline. The uGP model has the capability to absorb the location uncertainty into the covariance function. Due to this flexibility, it can handle higher values of \( \lambda \) and still maintain an almost constant \( \hat{d}_c \) and \( \hat{\sigma}_q \) with increase in \( \lambda \). For fair comparison with cGP, we also consider the case where \( \sigma_{\text{proc}} \) is estimated as part of the learning, referred to as uGP-proc. It can be observed in Fig. 4(c) that \( \hat{\sigma}_{\text{proc}} \) increases with increase in \( \lambda \). When comparing uGP-proc to uGP, we observe a lower value of \( \hat{\sigma}_q \) and a higher value of \( \hat{d}_c \) for a particular value of \( \lambda \). From this, we conclude that uGP should be preferred over uGP-proc, as it leads to simpler optimization. Finally, note that the error bars of the uGP estimates are relatively small when compared to cGP.

C. Prediction

---

Four cases can be considered, depending on whether training or testing inputs are in \( X \) or \( U \). We will focus on the case where either training or test locations are uncertain, but not both. From these, the behavior when both training and testing inputs are in \( U \) can be easily understood.

- **Uncertain training locations and certain testing locations:** In this case \( u_i \in U \) and \( u_* \in X \). Fig. 5(a) depicts the prediction results in terms of the predictive mean and predictive standard deviation (shown as shaded areas) for a particular realization of the channel field. It can be observed that uGP is able to predict the received power comparatively better than cGP and MCGP. uGP is able to estimate the underlying channel parameters better with the expected covariance function, which takes in to account the location uncertainty of the nodes. In turn, this means that uGP can track the faster variations in the channel. cGP tries to model the true function with a slow varying process due to very high \( \hat{d}_c \). Furthermore, cGP has higher uncertainty in predictions due to high \( \hat{\sigma}_{\text{proc}} \) (see Fig. 4(c)). On the other hand, MCGP has slightly better prediction performance compared to cGP due to the averaging by drawing samples from the distribution of the uncertain training locations. Averaging the prediction error over multiple channel realizations, Fig. 5(b) shows the mean squared error (MSE) of the received power prediction of cGP and uGP with respect to \( \lambda \) (MCGP is not shown due to its poor learning performance). uGP clearly outperforms cGP (except \( \lambda = 0 \)) due to its better tracking of the true channel (see Fig. 5(a)) despite uncertainty on the training locations. The reason for higher MSE in the case of \( \lambda = 0 \) for uGP is due to its kernel mismatch.

- **Certain training locations and uncertain testing locations:** In this case \( u_i \in X \) and \( u_* \in U \) (with a constant location error standard deviation \( \sigma \)). Now the performance must be assessed with respect to the expected received power \( P_{RX,\text{avg}}(u_*) = \int P_{RX}(x_*) p(x_*) dx_* \), where \( p(x_*) = N(z_*, \sigma^2 I) \), in which \( z_* \) is the mean of distribution described by \( u_* \). An example is shown in Fig. 6(a), depicting \( P_{RX,\text{avg}} \) as a function of \( z_* \), as well as the predictions from cGP, MCGP, and uGP. It can be observed that uGP and MCGP follow well \( P_{RX,\text{avg}} \). Specifically, MCGP tracks \( P_{RX,\text{avg}} \) quite closely as it is optimal in this case. In contrast, cGP follows the actual received power at \( z_* \), rather than the averaged power. This leads to fast variations in cGP, which are not present in uGP and MCGP. Fig. 6(b) shows the MSE of the received power prediction of cGP, MCGP, and uGP with respect to \( \sigma \) when averaging the prediction error over multiple channel realizations. As expected, MCGP has the lower MSE than uGP and cGP. However, uGP performs better than cGP for all considered \( \sigma \), except \( \sigma = 0 \) (due to kernel mismatch). Furthermore, the performance of uGP is very close to that of MCGP.
D. Resource Allocation Example

1) Scenario: In this section, we compare cGP and uGP for a simple proactive resource allocation scenario. We consider a user moving through a region \( \mathcal{A} \) and predict the CQM at each location. The supported rate, expressed in bits per channel use (bpu), for a user at location \( x_* \), is defined as

\[
r(x_*) = \log_2 \left( 1 + \text{SNR}(x_*) \right),
\]

where \( \text{SNR}(x_*) = \frac{P_{\text{RX}}(x_*)}{W_{\text{lin}}} \) is the signal-to-noise ratio at location \( x_* \), \( W_{\text{lin}} \) is the receiver thermal noise and \( P_{\text{RX}}(x_*) \) is the received power, both measured in linear scale. The average rate in the region \( \mathcal{A} \), denoted as \( \bar{r}_\mathcal{A}^{\text{ref}} \), is defined as

\[
\bar{r}_\mathcal{A}^{\text{ref}} = \frac{1}{|\mathcal{A}|} \int_\mathcal{A} r(x_*) \mathrm{d}x_*,
\]

where \( |\mathcal{A}| \) denotes the area of the region \( \mathcal{A} \). The predicted rate for a user at a future location \( x_* \), based on the predicted CQM values \( \bar{P}_{\text{RX}}(x_*), \tilde{P}_{\text{RX}}(x_*) \), is defined as

\[
r(x_*, \alpha) = \log_2 \left( 1 + \text{SNR}(x_*, \alpha) \right),
\]

where \( \alpha \geq 0 \) is a confidence parameter, \( \text{SNR}(x_*, \alpha) = \frac{P_{\text{RX}}(x_*, \alpha)}{W_{\text{lin}}} \) and \( P_{\text{RX}}(x_*, \alpha) = 10 \log_{10} \left( \frac{P_{\text{RX}}(x_*, \alpha)}{W_{\text{lin}}} \right) = \bar{P}_{\text{RX}}(x_*) - \alpha \left( \tilde{P}_{\text{RX}}(x_*) \right)^\alpha \).

Figure 5. Performance comparison of cGP, MCGP, and uGP under uncertain training and certain testing locations. Inset (a) received power prediction using uncertain training locations with average location error of \( \lambda = 8 \) m and certain test locations for single realization of a channel field. The shaded area (grey for cGP and blue for UGP) depicts point-wise predictive mean plus and minus the predictive standard deviation, and (b) MSE performance of cGP and uGP as a function of average location error standard deviation \( \lambda \). The MSE is averaged for each value of \( \lambda \) and for 50 realizations of the channel field is shown as the mean of the MSE and error bars with one standard deviation. The MSE is calculated as \( \frac{1}{|\mathcal{T}|} \sum_{x_*, \alpha \in \mathcal{T}} (\bar{P}_{\text{RX}}(x_*) - P_{\text{RX}}(x_*))^2 \), where \( \mathcal{T} \) is the set of test locations and \( |\mathcal{T}| \) denotes its cardinality.

Figure 6. Performance comparison of cGP, MCGP, and uGP under certain training and uncertain testing locations. Inset (a) received power prediction using certain training and uncertain test locations with a constant location error standard deviation \( \sigma = 5 \) m for single realization of channel field, and (b) MSE performance of cGP, MCGP and uGP as a function constant location error standard deviation \( \sigma \) on test locations. The MSE is averaged for each value of \( \sigma \) and for 50 realizations of the channel field is shown as the mean of the MSE and error bars with one standard deviation. The MSE is calculated as \( \frac{1}{|\mathcal{T}^{\mu}|} \sum_{x_*, \alpha \in \mathcal{T}^{\mu}} (\bar{P}_{\text{RX}}(x_*) - P_{\text{RX}}(x_*))^2 \), where \( \mathcal{T}^{\mu} \) is the set of test location distributions and \( |\mathcal{T}^{\mu}| \) denotes its cardinality.
2) Performance Measure: The user moves through the environment according to a known trajectory. The base station allocates bits to each future location, proportional to \( r(x_\star, \alpha) \). When the user is at location \( x_\star \), only a fraction of the bits, proportional to \( \min(r(x_\star, \alpha), r(x_\star)) \) would be delivered. Therefore, the effective rate \( r_{\text{eff}}(x_\star, \alpha) \) for the user at location \( x_\star \) is

\[
    r_{\text{eff}}(x_\star, \alpha) = \min(r(x_\star, \alpha), r(x_\star)).
\]

The average effective rate \( \bar{r}_{\text{eff}}(\alpha) \) for a given confidence level \( \alpha \) is then computed by spatial average of \( r_{\text{eff}}(x_\star, \alpha) \) over region \( \mathcal{A} \) as

\[
    \bar{r}_{\text{eff}}(\alpha) = \frac{1}{|\mathcal{A}|} \int_{\mathcal{A}} r_{\text{eff}}(x_\star, \alpha) \, dx_\star \in [0, \bar{r}_{\text{eff}}].
\]

When \( r(x_\star, \alpha) > r(x_\star) \), a part of the allocated bits cannot be delivered. The total fraction of undelivered bits over the environment is given by

\[
    U(\alpha) = \frac{1}{|\mathcal{A}|} \int_{\mathcal{A}} (r(x_\star, \alpha) - r_{\text{eff}}(x_\star, \alpha)) dx_\star \in [0, 1].
\]

Hence, \( \bar{r}_{\text{eff}}(\alpha) \) describes the rate that the user will receive (penalizing under-estimation of the rate), while \( U(\alpha) \) describes the loss due to lost bits (penalizing over-estimating of the rate).

3) Predicted communication rates with uncertain training locations: We predict the CQM based on training with uncertain locations (considering \( \lambda \in \{0, 10\} \) m) at known test locations \( x_\star \in \mathcal{X} \) within a one-dimensional region \( \mathcal{A} \). The average effective rate \( \bar{r}_{\text{eff}}(\alpha) \) and the fraction of undelivered bits \( U(\alpha) \), as a function of \( \alpha \), are shown in Fig 7(a)–(b), respectively. As expected, increasing \( \alpha \) leads to a more conservative allocation, thus reducing both \( \bar{r}_{\text{eff}}(\alpha) \) and \( U(\alpha) \).

For a specific value of \( \alpha \), increase in \( \lambda \) decreases \( \bar{r}_{\text{eff}}(\alpha) \). This is due to the fact that with increase in \( \lambda \), the mean \( \mu_{\text{signal}}(x_\star) \) is poor and the variance \( \sigma_{\text{signal}}(x_\star) \) is high for CQM predictions.

It is evident that when \( \lambda = 0 \), uGP and cGP attain similar performance, both in terms of \( \bar{r}_{\text{eff}}(\alpha) \) and \( U(\alpha) \). When \( \lambda \) is increased to 10 m, cGP suffers from a significant reduction in effective rate \( \bar{r}_{\text{eff}}(\alpha) \), while at the same time dropping up to 4.5% of the bits. This is due to cGP’s poor predictions, which are either too low (leading to a reduction in \( \bar{r}_{\text{eff}}(\alpha) \)) or too high (leading to an increase in \( U(\alpha) \)). In contrast, uGP, which is able to track the channel well despite uncertain training, achieves a higher effective rate, especially for high confidence values (e.g., around 2 times higher rate for \( \alpha = 3 \), for \( U(\alpha) \) less than 0.1%).

VII. CONCLUSION

Channel quality metrics can be predicted using spatial regression tools such as Gaussian processes (GP). We have studied the impact of location uncertainties on GP and have demonstrated that, when heterogeneous location uncertainties are present, the classical GP framework is unable to (i) learn the underlying channel parameters properly; (ii) predict the expected channel quality metric. By introducing a GP that operates directly on the location distribution, we find uncertain GP (uGP), which is able to both learn and predict in the presence of location uncertainties. This translates in better performance when using uGP for predictive resource allocation.

Possible avenues of future research include modeling correlation of shadowing in the temporal dimension, study of better approximations for learning with uncertain locations, and the extension to ad-hoc networks.

APPENDIX A

APPROXIMATION OF EXPECTED MEAN FUNCTION

Let \( d_i = ||x_i|| \) and recall from random variable transformation theory that

\[
    \int \log_{10}(||x_i||) p(x_i) \, dx_i = \int \log_{10}(d_i) p(d_i) \, dd_i.
\]

Figure 7. Resource allocation example for cGP, and uGP with two different values of localization error standard deviations (\( \lambda \in \{0, 10\} \) m) and for different values of the confidence parameter \( \alpha \). The results are averaged for each value of \( \lambda \) with 50 channel realizations. Inset (a) the effective rate \( \bar{r}_{\text{eff}}(\alpha) \), and (b) the fraction of undelivered bits \( U(\alpha) \).
We assume \( p(x_i) = \mathcal{N}(z_i, \sigma_i^2 \mathbf{I}) \), so \( p(d_i) \) follows a Rician distribution
\[
p(d_i) = \frac{d_i}{\sigma_i^2} \exp\left(-\frac{d_i^2}{2 \sigma_i^2}\right) I_0\left(\frac{\|z_i\| d_i}{\sigma_i^2}\right) d_i > 0, \tag{40}\]
where \( I_0(.) \) is a modified Bessel function of zero-th order. For \( \|z_i\|/\sigma_i \geq 3 \), \( p(d_i) \) can be approximated as a Gaussian distribution
\[
p_{\text{Gauss}}(d_i) = \frac{1}{\sqrt{2 \pi \sigma_i^2}} \exp\left(-\frac{(\|z_i\| - d_i)^2}{2 \sigma_i^2}\right). \tag{41}\]

The integral \([39]\) still does not have a closed form expression with \( p_{\text{Gauss}}(d_i) \). Now approximating the \( \log_{0.1}(.) \) function with a polynomial function of the form \( w(d_i) = \sum_{j=0}^{J} a_j d_i^j \) then \([39]\) can be written as
\[
\int \log_{0.1}(\|x_i\|) p(x_i) \, dx_i \approx \int_{-\infty}^{+\infty} w(d_i) p_{\text{Gauss}}(d_i) \, dd_i, \tag{42}\]
which can be computed exactly.

Appendix B
Learning procedure

In this appendix, we show the learning of the hyperparameters \( \Theta = [\sigma_n, \sigma_{\text{proc}}, d_c, L_0, \eta, \sigma_\Psi] \) for cGP, uGP, and MCGP. We consider nodes know \( \sigma_n \) and \( L_0 \) from their measurements, therefore they are not estimated as part of the learning process. Let the remaining set of hyperparameters be \( \theta = [\sigma_{\text{proc}}, d_c, \sigma_\Psi] \) and \( \eta \).

cGP

Based on Section III, we can write the received measurements \( y \) with their corresponding training locations \( x \) in matrix form as
\[
y = 1^T L_0 + h_x \eta + \Psi + n, \tag{43}\]
where \( \Psi = [\Psi(x_1), \ldots, \Psi(x_N)]^T, n = [n_1, \ldots, n_N]^T \), and \( h_x = -10 \{\log_{10}(\|x_1\|), \ldots, \log_{10}(\|x_N\|)\}^T \). Assuming the measurements are uncorrelated, then the least squares estimate of the path-loss exponent can be computed as
\[
\hat{\eta} = (h_x^T h_x)^{-1} h_x^T (y - 1^T L_0). \tag{44}\]

Once the path-loss exponent is estimated, the mean component of the received measurements can be subtracted as, \( y_c = y - 1^T L_0 - h_x \hat{\eta} \). Then, \( y_c \) becomes a zero-mean Gaussian process. The likelihood function \( l(\theta) \) becomes
\[
l(\theta) = p(y_c | x, \theta) = \mathcal{N}(y_c, 0, K). \tag{45}\]

The hyperparameters \( \theta \) are estimated by minimizing negative logarithm of \( l(\theta) \)
\[
\hat{\theta} = \arg \min_{\theta} \left\{ -\log(p(y_c | x, \theta)) \right\} = \arg \min_{\theta} \left\{ \log |K| + \frac{1}{2} y_c^T K^{-1} y_c \right\}. \tag{46}\]

We calculate the variance of the process \( y_c \) as
\[
\sigma_{\text{tot}}^2 = 1/N \sum_{i=1}^{N} \|y_c_i\|^2. \tag{47}\]

uGP

In this case, the path-loss exponent is estimated as
\[
\hat{\eta} = (h_u^T h_u)^{-1} h_u^T (y - 1^T L_0), \tag{48}\]
where \( h_u = -10 \{\log_{10}(\|x_1\|), \ldots, \log_{10}(\|x_N\|)\}^T \). Once again removing the mean from the measurements, we obtain \( y_u = y - 1^T L_0 - h_u \hat{\eta} \). The hyperparameters \( \theta \) are estimated by minimizing the modified negative log-likelihood function
\[
\hat{\theta} = \arg \min_{\theta} \left\{ -\log(p(y_u | U, \theta)) \right\} = \arg \min_{\theta} \left\{ \log |K_u| + \frac{1}{2} y_u^T K_u^{-1} y_u \right\}. \tag{49}\]

MCGP

It is no longer feasible to estimate \( \eta \) first and subtract to make the process zero mean, because of summation in the Monte Carlo integration \([12]\). Therefore, we optimize \([13]\) with respect to the hyperparameters \( \eta \) and \( \theta \) using \textit{fminsearch} function of Matlab.

ACKNOWLEDGMENT

The authors would like to thank Ido Nevat and Lennart Svensson for their feedback on the manuscript.

REFERENCES

[1] S. Sand, R. Tanbourgi, C. Mensing, and R. Raulefs, “Position aware adaptive communication systems,” in Forty-Third Asilomar Conference on Signals, Systems and Computers, 2009, pp. 73–77.

[2] R. Di Taranto, S. Muppireddy, R. Raulefs, D. Slock, T. Svensson, and H. Wymeersch, “Location-aware communications for 5G networks,” IEEE Signal Processing Magazine, vol. 31, no. 6, pp. 102–112, Nov 2014.

[3] H. Abou-zeid, H. Hassanein, and S. Valentín, “Optimal predictive resource allocation: Exploiting mobility patterns and radio maps,” in IEEE Globecom Workshops, Dec 2013, pp. 4877–4882.

[4] J. Johansson, W. Hapsari, S. Kelley, and G. Bodog, “Minimization of drive tests in 3GPP release 11,” IEEE Communications Magazine, vol. 50, no. 11, pp. 36–43, 2012.

[5] A. Goldsmith, Wireless communications. Cambridge university press, 2005.

[6] N. Jalden, P. Zetterberg, B. Ottersten, A. Hong, and R. Thoma, “Correlation properties of large scale fading based on indoor measurements,” IEEE Communications Magazine, vol. 49, no. 11, pp. 30–39, 2011.

[7] M. S. Grewal, L. R. Weill, and A. P. Andrews, Global positioning systems, inertial navigation, and integration. John Wiley & Sons, 2001.

[8] C. Rasmussen and C. Williams, Gaussian processes for machine learning. MIT Press, 2006.

\( ^4 \)Since \( \sigma_{\text{proc}} \) only captures kernel mismatch irrespective of the location uncertainty, the value of \( \sigma_{\text{proc}} \) can be obtained with noise-free training locations by performing learning as in the case of cGP but with a covariance function of the form \( \text{(4)} \) for \( p = 2 \).
[9] J. Quiñonero-Candela and C. E. Rasmussen, “A unifying view of sparse approximate Gaussian process regression,” *The Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.

[10] L. Csató and M. Opper, “Sparse on-line Gaussian processes,” *Neural computation*, vol. 14, no. 3, pp. 641–668, 2002.

[11] E. Snelson and Z. Ghahramani, “Sparse Gaussian processes using pseudo-inputs,” *Advances in neural information processing systems*, vol. 18, p. 1257, 2006.

[12] S. Sarkka, A. Solin, and J. Hartikainen, “Spatiotemporal learning via infinite-dimensional bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering,” *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 51–61, 2013.

[13] P. Dallaire, C. Besse, and B. Chaib-draa, “An approximate inference with Gaussian process to latent functions from uncertain data,” *Neurocomputing*, vol. 74, no. 11, pp. 1945–1955, 2011.

[14] A. Girard, “Approximate methods for propagation of uncertainty with Gaussian process models,” Ph.D. dissertation, University of Glasgow, 2004.

[15] J. Fink, “Communication for teams of networked robots,” Ph.D. dissertation, Elect. Syst. Eng., Univ. Pennsylvania, Philadelphia, PA, Aug. 2011.

[16] P. Agrawal and N. Patwari, “Correlated link shadow fading in multi-hop wireless networks,” *IEEE Transactions on Wireless Communications*, vol. 8, no. 8, pp. 4024–4036, 2009.

[17] S.-J. Kim, E. Dall’Anese, and G. Giannakis, “Cooperative spectrum sensing for cognitive radios using kriged Kalman filtering,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 1, pp. 24–36, 2011.

[18] M. Malmirchegini and Y. Mostofi, “On the spatial predictability of communication channels,” *IEEE Transactions on Wireless Communications*, vol. 11, no. 3, pp. 964–978, 2012.

[19] Y. Yan and Y. Mostofi, “Impact of localization errors on wireless channel prediction in mobile robotic networks,” in *IEEE Globecom, Workshop on Wireless Networking for Unmanned Autonomous Vehicles*, Dec. 2013.

[20] M. Jadaílíha, Y. Xu, J. Choi, N. Johnson, and W. Li, “Gaussian process regression for sensor networks under localization uncertainty,” *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 223–237, 2013.

[21] S. S. Szyszcowicz, H. Yanikomeroglu, and J. S. Thompson, “On the feasibility of wireless shadowing correlation models,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 9, pp. 4222–4236, 2010.

[22] M. Gudmundson, “Correlation model for shadow fading in mobile radio systems,” *Electronics letters*, vol. 27, no. 23, pp. 2145–2146, 1991.

[23] A. McHutchon and C. E. Rasmussen, “Gaussian process training with input noise,” in *Advances in Neural Information Processing Systems*, 2011, pp. 1341–1349.

[24] A. Girard and R. Murray-Smith, “Learning a Gaussian process model with uncertain inputs,” Department of Computing Science, University of Glasgow, Tech. Rep. TR-2003-144, June 2003.