IEEE Open Journal of
Antennas and Propagation

Received 18 March 2020; revised 17 April 2020; accepted 19 April 2020. Date of publication 23 April 2020; date of current version 18 May 2020.

Digital Object Identifier 10.1109/OJAP.2020.2989814

Learning Parameters of Stochastic Radio Channel Models From Summaries

AYUSH BHARTI (Graduate Student Member, IEEE), RAMONI ADEOGUN (Senior Member, IEEE), AND TROELS PEDERSEN (Member, IEEE)

Department of Electronic Systems, Aalborg University, 9220 Aalborg, Denmark

CORRESPONDING AUTHOR: A. BHARTI (e-mail: ayb@es.aau.dk)

This work is supported by the Danish Council for Independent Research, under Grant DFF 7017-00265 and performed within the framework of the COST Action CA15104 IRACON.

ABSTRACT Estimating parameters of stochastic radio channel models based on new measurement data is an arduous task usually involving multiple steps such as multipath extraction and clustering. We propose two different machine learning methods, one based on approximate Bayesian computation (ABC) and the other on deep learning, for fitting stochastic channel models to data directly. The proposed methods make use of easy-to-compute summary statistics of measured data instead of relying on extracted multipath components. Moreover, the need for post-processing of the extracted multipath components is omitted. Taking the polarimetric propagation graph model as an example stochastic model, we present relevant summaries and evaluate the performance of the proposed methods on simulated and measured data. We find that the methods are able to learn the parameters of the model accurately in simulations. Applying the methods on 60 GHz indoor measurement data yields parameter estimates that generate averaged power delay profile from the model that fits the data.

INDEX TERMS Machine learning, Monte Carlo methods, deep learning, Bayesian inference, radio channel modeling, approximate Bayesian computation, summary statistics, propagation graph, parameter estimation, likelihood.

I. INTRODUCTION

STOCHASTIC models of the radio channel are indispensable tools in the design and analysis of communication and localization systems. Stochastic radio channel models are used for characterizing and simulating realizations of the channel in different environments. However, for the model to generate data similar to what is observed in the measurements, its parameters need to be learned from the data. The process of learning or estimating the parameters of a model from new measurements is termed as calibration. Calibration could be obtained by deriving the parameters theoretically, e.g., in room electromagnetics or in ray tracing. In fact, some parameters such as speed of light or room geometry are set not using the data. Standard calibration technique using data would be to either maximize the likelihood function of the data with respect to the parameters, or to characterize the posterior distribution of the parameters in a Bayesian sense. However, most stochastic channel models suffer from intractability of the likelihood function, and therefore, calibrating them given a new set of measurement data is challenging [1].

Typically, stochastic multipath radio channel models are calibrated in steps, as described in Fig. 1(a). This calibration methodology has been followed since the early works of Turin *et al.* [2] and Saleh and Valenzuela [3] till more recent stochastic channel models [4]–[8]. First, the data is reduced to a set of multipath components, each having their own gain, delay, etc., by applying high-resolution algorithms such as SAGE (Space Alternating Generalized Expectation-maximization) [9], MUSIC (MUltiple SIgnal Classification), ESPRIT (Estimation of Signal Parameters via Rotational Invariance Techniques), and RiMAX [10], among others. The extracted multipath components are then used to estimate the model parameters. In case of cluster-based channel models, an additional clustering step can be applied. Alternatively, the presence of clusters, or other multipath effects, can be included in the derivation of the high-resolution estimator, as in [11]–[13], to obtain the cluster parameters directly. In

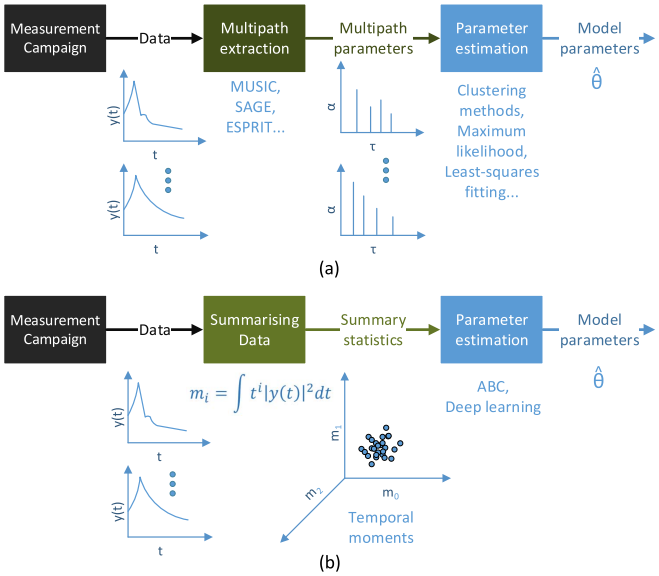

FIGURE 1. State-of-the-art calibration methodology (a), versus the proposed method (b).

a final step, the model parameters are estimated from the extracted multipath components and clusters.

Even though such multi-step calibration approaches are widely used, they suffer from a range of issues. It can be cumbersome and labor-intensive to derive, implement, and test sophisticated multipath extraction and clustering algorithms that require a number of heuristic choices to be made. Moreover, the estimation of the multipath components is prone to errors due to censoring [14]. Therefore, the overall estimation accuracy of the model parameters is difficult to determine due to this step-by-step calibration approach.

There may exist statistics other than the multipath components that are easier to obtain, and still hold enough information to be able to learn the model parameters. Potentially, the parameters of the channel models can be estimated without the multipath extraction step by relying on these easy-to-compute summary statistics. Such an estimator for the Saleh and Valenzuela model [3] was proposed in [15] where the estimation problem was framed as an optimization problem that fitted summary statistics of the data with approximate analytical expressions. More recently, multipath extraction-free calibration methods based on sampling [16] and method of moments [17] have been developed and applied to the Turin model. These methods summarize the data into certain statistics, and rely on explicit derivation of equations linking their means and covariances to the model parameters. Drawback of these methods is that such equations need to be derived for every stochastic channel model, which is either non-trivial or oftentimes not possible.

In the present contribution, we further advance the idea of using other summaries than the multipath parameters for model calibration. This leads to the calibration methodology outlined in Fig. 1(b) where the data is first summarized into a set of statistics from which the model parameters are obtained. We extend our previous work on learning parameters of stochastic channel models using approximate Bayesian computation (ABC) [18] and deep learning (DL) [19], where we had applied the methods on the cluster-based model of Saleh and Valenzuela [3].

ABC is a framework for performing likelihood-free inference on generative models with intractable likelihoods [20], [21], as is the case for stochastic channel models. It relies on sampling parameter values from a prior distribution, simulating data from the model, and comparing the simulated summaries to those obtained from the measurements. Parameter values that yield summaries similar to the observed ones are used to approximate the posterior distribution. Initially developed in the field of population genetics [22], ABC has since been applied in various other fields of research such as ecology [23], astrophysics [24], and structural dynamics [25], to name a few. To the best of our knowledge, ABC has not been applied in wireless communications except for our previous conference paper [18].

The DL method utilizes a neural network (NN) to establish a functional relationship between the summaries and the parameters, and uses the trained NN to estimate the parameters given the observed summaries. Neural networks have been proven to exhibit capability for universal approximation of any continuous real-valued function [26]. These networks have been successfully applied in fields such as computer vision and image processing over the last several years. Recently, the wireless communications community has also explored avenues for application of DL. While efforts have been directed towards DL-enabled physical layer design, only a few applications to radio channel modeling and calibration have been proposed. In [27], the authors utilized Deep Neural Network for uplink-downlink channel calibration in massive MIMO. Similar network is utilized for predicting path-loss exponent from millimeter wave channel measurements in [28]. The DL method proposed here is a generalization of the framework introduced in [19] where a single layer neural network is applied to estimate parameters of propagation models. Similar DL-based likelihood-free inference framework has been applied in population genetics [29].

In this paper, we present two machine learning methods based on ABC and DL to calibrate stochastic radio channel models without multipath extraction. We show the applicability of the methods by calibrating the polarimetric propagation graph (PG) model [30] as an example, since multipath extraction cannot be directly applied to calibrate it. We also present a number of summary statistics for representing channel measurements that are used as input for the learning methods. The chosen statistics are qualified via simulation study of the PG model. Simulation results illustrate the capability of the proposed learning methods to accurately estimate the model parameters. The methods are also applied to calibrate the PG model using real indoor channel measurements. Reasonable fits of the averaged power delay profile were seen between the measurements and the model, thus validating the proposed learning methods.

The remainder of the article is organized as follows. We describe the ABC and DL calibration methods in Section II. The description of the polarimetric PG model is given in Section III, along with the choice of summary statistics. In Section IV, we apply the ABC and DL methods on the PG model and show the results. The discussion regarding the proposed methods is presented in Section V, and Section VI provides our concluding remarks.

II. SUMMARY-BASED CALIBRATION METHODS

Consider a stochastic generative model, $\mathcal{M}(\boldsymbol{\theta})$, which is easy to simulate from given any value of the parameter, $\boldsymbol{\theta}$. Each time the model is called for a given parameter vector, it generates independent realizations of simulated data, $\mathbf{Y}$. Here, $\mathbf{Y}$ could be a vector or a matrix. Let $\mathbf{Y}_{\text{obs}}$ be a set of measurement data obtained experimentally. The calibration problem then involves estimating $\boldsymbol{\theta}$ such that the model, $\mathcal{M}(\boldsymbol{\theta})$, fits to the measured data, $\mathbf{Y}_{\text{obs}}$. However, the likelihood function of $\mathbf{Y}$ given $\boldsymbol{\theta}$ is intractable, and so standard estimation techniques are not applicable. Typically, $\mathbf{Y}$ is a high-dimensional data matrix, as is the case with radio channel transfer function or impulse response measurements for multiple independent realizations of the channel. Therefore, we use a function, $S(\cdot)$, that summarizes $\mathbf{Y}$ into a set of q statistics, $\mathbf{s} \in \mathbb{R}^q$, such that $\mathbf{s} = S(\mathbf{Y})$. We then use these statistics as data in our calibration methods to estimate $\boldsymbol{\theta}$ given the observations $\mathbf{s}_{\text{obs}} = S(\mathbf{Y}_{\text{obs}})$. Ideal choice for $\mathbf{s}$ would be sufficient statistics of $\mathbf{Y}$, but those are unavailable in most practical cases.

A. APPROXIMATE BAYESIAN COMPUTATION

ABC is a likelihood-free inference method that samples from the approximate posterior distribution of the parameters by finding values that lead to simulated datasets from the model that are similar to the observed data. The method involves sampling from the prior distribution of the parameters, $p(\boldsymbol{\theta})$, and then generating datasets from the model. These simulated datasets are then compared to the observed set of measurements in some distance metric, $\rho(\cdot, \cdot)$, and the values of $\boldsymbol{\theta}$ that yield a distance smaller than a pre-defined tolerance threshold, ϵ, form samples from the approximate posterior distribution. These samples can then be used to approximate standard point estimates of $\boldsymbol{\theta}$ such as the minimum mean square error (MMSE) or the maximum *a posteriori* estimate. The basic rejection-ABC algorithm can be summarized as follows:

1) Sample from prior, $\boldsymbol{\theta}^* \sim p(\boldsymbol{\theta})$
2) Simulate $\mathbf{Y}^* \sim \mathcal{M}(\boldsymbol{\theta}^*)$
3) Compute summary statistics $\mathbf{s}^* = S(\mathbf{Y}^*)$
4) Accept $\boldsymbol{\theta}^*$ if $\rho(\mathbf{s}^*, \mathbf{s}_{\text{obs}}) < \epsilon$

The approximation here arises on account of summarizing the data into a set of statistics, and accepting samples within a tolerance threshold. Choosing sufficient statistics to be used in ABC mitigates the former approximation, however, finding sufficient statistics is typically not feasible. Choosing a small ϵ improves the posterior approximation but increases the rejection rate significantly, whereas a large ϵ leads to sampling from almost the prior. Therefore, the distance metric, the tolerance threshold, and the summary statistics are the necessary ingredients required to implement an ABC algorithm.

The choice of appropriate summary statistics is crucial to the quality of approximation [31], and it depends upon the application and the model at hand. Given a pool of summary statistics, there exist statistical methods to, e.g., select best subset among them, or construct a much smaller set of highly informative statistics through projection techniques, among others [32]. However, domain knowledge is vital for constructing appropriate summaries that are not only informative about the model parameters, but also relevant from an application perspective. Thus, domain experts can choose meaningful summaries that they wish to fit to the model. The specific statistics used in this paper will be addressed in Section III-D with regards to a specific example radio channel model.

Typically, the Euclidean distance between the observed and the simulated statistics is used as the distance metric in summary-based ABC methods, i.e., $\rho(\cdot, \cdot) = \|\cdot\|$. Other distance metrics can also be used, however, the Euclidean distance seems appropriate in the context of summary-based ABC methods for the application at hand. In cases where a set of statistics are used that differ in their units and order of magnitude, it is important to normalize them before computing the distance [33]. The normalization aims to bring all the individual distances to the same scale, such that the total distance will be their sum. In this paper, we take $\rho(\cdot, \cdot)$ to be the Euclidean distance between the normalized statistics, but still denote it as $\|\mathbf{s} - \mathbf{s}_{\text{obs}}\|$, with the normalization of the statistics assumed to be implicit.

Specifying an appropriate value of ϵ in terms of the distance may prove to be difficult. Setting ϵ too low leads to unknown run time of the algorithm to get a certain number of accepted samples. Therefore, it is usual to employ a k-nearest neighbor approach and specify ϵ as a percentile of the total simulated samples. That is, out of M samples of $\boldsymbol{\theta}$ from the prior, we accept the first $M_\epsilon = \epsilon M$ samples leading to the smallest $\|\mathbf{s} - \mathbf{s}_{\text{obs}}\|$.

The basic ABC method is simple but can be rather slow. Instead, we propose to use a sequential sampling method, specifically the Population Monte Carlo ABC (PMC-ABC) [34], and supplement it by employing the local-linear regression adjustment method proposed in [31]. The resulting algorithm, named PMC-ABC with regression adjustment, is detailed in Alg. 2. A block diagram summarizing the proposed method is shown in Fig. 2. In the following, we describe the two ABC techniques.

1) REGRESSION ABC

We supplement the rejection-ABC algorithm by employing the local-linear regression adjustment method proposed in [31]. The regression adjustment improves the posterior

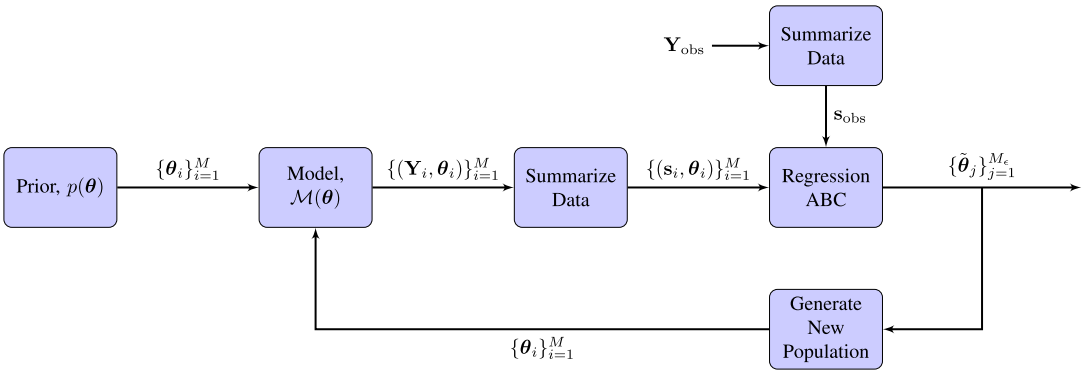

FIGURE 2. Block diagram depicting data flow in the proposed PMC-ABC algorithm with regression adjustment.

approximation by 1) weighting the accepted parameter samples according to their corresponding distance value and 2) adjusting them using a linear regression model applied locally in the vicinity of $\mathbf{s}_{\text{obs}}$. For $\boldsymbol{\theta}_j$ being the j^{th} accepted parameter sample and $\mathbf{s}_j$ the corresponding simulated statistics vector, the linear model reads

$$\boldsymbol{\theta}_j = \boldsymbol{\alpha} + \left(\mathbf{s}_j - \mathbf{s}_{\text{obs}}\right)^T \boldsymbol{\beta} + \boldsymbol{\varepsilon}_j, \quad j = 1, \ldots, M_\epsilon, \tag{1}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^p$ and $\boldsymbol{\beta} \in \mathbb{R}^{q \times p}$ are the intercept and regression coefficients, respectively, and $\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_{M_\epsilon}$ are uncorrelated noise variables with zero mean. The least-squares estimate of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are obtained by solving the optimization problem

$$\underset{\boldsymbol{\alpha},\boldsymbol{\beta}}{\text{argmin}} \sum_{j=1}^{M_\epsilon} \left[\boldsymbol{\theta}_j - \boldsymbol{\alpha} - \left(\mathbf{s}_j - \mathbf{s}_{\text{obs}}\right)^T \boldsymbol{\beta}\right]^2 \mathcal{K}_\epsilon\left(||\mathbf{s}_j - \mathbf{s}_{\text{obs}}||\right). \tag{2}$$

The Epanechnikov kernel, $\mathcal{K}_\epsilon(\cdot)$, depends on the maximum accepted distance based on the chosen ϵ, and ensures that the regression model is applied locally. The samples are then adjusted as

$$\tilde{\boldsymbol{\theta}}_j = \boldsymbol{\theta}_j - \left(\mathbf{s}_j - \mathbf{s}_{\text{obs}}\right)^T \hat{\boldsymbol{\beta}}, \quad j = 1, \ldots, M_\epsilon, \tag{3}$$

thus improving the approximation to the posterior distribution. Note that the adjustment is applied on each entry of the parameter vector independently. This regression-ABC algorithm is described in Alg. 1.

The adjustment in (3) is done disregarding the prior range of the parameters. Therefore, the regression method may adjust the samples to fall outside the support of the prior. This issue can be addressed by transforming the parameters before adjustment [31]. A log transformation can be used for positive parameters, and a logit transformation for parameters with bounded priors [35].

2) POPULATION MONTE CARLO ABC

In cases where the parameter vector is high-dimensional, rejection-ABC method needs a large number of simulations of the model to reasonably explore the parameter space. Therefore, more advanced ABC methods have been introduced that rely on Markov chain Monte Carlo and Sequential Monte Carlo techniques that sample the parameter space

Algorithm 1 Regression ABC Algorithm

Input: Parameter values $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M)$ and corresponding simulated statistics $(\mathbf{s}_1, \ldots, \mathbf{s}_M)$, observed statistics $\mathbf{s}_{\text{obs}}$, number of accepted samples M_ϵ,

Accept $(\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_{M_\epsilon}^*) \sim \{\boldsymbol{\theta}_i\}_{i=1}^{M}$ with the smallest $\|\mathbf{s}_i - \mathbf{s}_{\text{obs}}\|$

Solve optimisation problem (2) with $\left\{\theta_j^*\right\}_{j=1}^{M_\epsilon}$ and corresponding $\left\{\mathbf{s}_j^*\right\}_{j=1}^{M_\epsilon}$ to get $\hat{\boldsymbol{\beta}}$

Adjust accepted samples $\left\{\boldsymbol{\theta}_j^*\right\}_{j=1}^{M_\epsilon}$ using (3) to get $\left\{\tilde{\boldsymbol{\theta}}_j\right\}_{j=1}^{M_\epsilon}$

Output: Samples $(\tilde{\boldsymbol{\theta}}_1, \ldots, \tilde{\boldsymbol{\theta}}_{M_\epsilon})$ from approximate posterior

efficiently. One such sequential technique is the Population Monte Carlo (PMC)-ABC method [34] that iteratively converges towards the approximate posterior distribution.

In the initialization of PMC-ABC, M_ϵ closest parameter samples out of $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M)$ are retained, similar to rejection-ABC. These accepted samples, $\{\boldsymbol{\theta}_j\}_{j=1}^{M_\epsilon}$, form an approximation to the posterior distribution. A new population of M parameter samples is then drawn from the density kernel

$$\varphi_t(\boldsymbol{\theta}) = \sum_{j=1}^{M_\epsilon} \mathbf{w}_j^{(t-1)} \mathcal{K}_t\left(\boldsymbol{\theta}^{(t)} | \boldsymbol{\theta}_j^{(t-1)}; \boldsymbol{\sigma}_{(t-1)}^2\right), \tag{4}$$

where t is the iteration index, $\mathbf{w}_j^{(t-1)}$ is the importance sampling weight associated with the accepted sample $\boldsymbol{\theta}_j^{(t-1)}$, and $\boldsymbol{\sigma}_{(t-1)}^2$ is a variance vector with each entry associated with a kernel $\mathcal{K}_t$. Note that $\mathbf{w}_j$ and $\boldsymbol{\sigma}^2$ are vectors of the same dimension as $\boldsymbol{\theta}$, and the new population for each parameter is drawn independently from the kernel with the corresponding variance. Typical choice for $\mathcal{K}_t$ is a Gaussian kernel, although other distributions may also be useful. A good choice for the variance of $\mathcal{K}_t$ is shown to be twice the empirical variance of the accepted samples [34]. Data and statistics are again simulated from the newly generated population, and the M_ϵ closest parameter samples are accepted and assigned weights $\mathbf{w}^{(t)} \propto p(\boldsymbol{\theta})/\varphi_t(\boldsymbol{\theta})$, where the division is taken entry-wise. This sequence of steps is repeated for T iterations, till the approximate posterior distributions converge.

Algorithm 2 PMC-ABC With Regression Adjustment

Input: Prior $p(\boldsymbol{\theta})$, model $\mathcal{M}(\boldsymbol{\theta})$, observed statistics $\mathbf{s}_{\text{obs}}$, M_ϵ, M, T

Initialization: $t = 1$,

for $i = 1$ to M **do**

Sample $\boldsymbol{\theta}_i^{(1)} \sim p(\boldsymbol{\theta})$

Simulate $\mathbf{Y}_i^{(1)} \sim \mathcal{M}(\boldsymbol{\theta}_i^{(1)})$ and compute $\mathbf{s}_i^{(1)} = S(\mathbf{Y}_i^{(1)})$

end for

Perform regression adjustment by applying **Algorithm 1** on $\left\{\left(\mathbf{s}_i^{(1)}, \boldsymbol{\theta}_i^{(1)}\right)\right\}_{i=1}^{M}$ to obtain $\left\{\tilde{\boldsymbol{\theta}}_j^{(1)}\right\}_{j=1}^{M_\epsilon}$

Set weights

$$\mathbf{w}_j^{(1)} = 1/M_\epsilon, \quad j = 1, \ldots, M_\epsilon, \quad \text{and variance}$$

$$\boldsymbol{\sigma}_{(1)}^2 = 2\widehat{\text{Var}}\left(\left\{\tilde{\boldsymbol{\theta}}_j^{(1)}\right\}_{j=1}^{M_\epsilon}\right)$$

for $t = 2$ to T **do**

for $i = 1, \ldots, M$ **do**

Sample $\boldsymbol{\theta}_i^* \sim \left\{\tilde{\boldsymbol{\theta}}_j^{(t-1)}\right\}_{j=1}^{M_\epsilon}$ with probabilities $\mathbf{w}_j^{(t-1)}$

Generate $\boldsymbol{\theta}_i^{(t)} \sim \mathcal{K}_t\left(\boldsymbol{\theta} | \boldsymbol{\theta}_i^*; \boldsymbol{\sigma}_{(t-1)}^2\right)$

Simulate $\mathbf{Y}_i^{(t)} \sim \mathcal{M}\left(\boldsymbol{\theta}_i^{(t)}\right)$ and compute $\mathbf{s}_i^{(t)} = S\left(\mathbf{Y}_i^{(t)}\right)$

end for

Perform regression adjustment by applying **Algorithm 1** on $\left\{\left(\mathbf{s}_i^{(t)}, \boldsymbol{\theta}_i^{(t)}\right)\right\}_{i=1}^{M}$ to obtain $\left\{\tilde{\boldsymbol{\theta}}_j^{(t)}\right\}_{j=1}^{M_\epsilon}$

Set weights

$$\mathbf{w}_j^{(t)} \propto \frac{p\left(\boldsymbol{\theta}_j^{(t)}\right)}{\sum_{j=1}^{M_\epsilon} \mathbf{w}_j^{(t-1)} \mathcal{K}_t\left(\boldsymbol{\theta}_j^{(t)} | \tilde{\boldsymbol{\theta}}_i^{(t-1)}; \boldsymbol{\sigma}_{(t-1)}^2\right)}, \quad 1 \le j \le M_\epsilon$$

$$\text{and variance} \quad \boldsymbol{\sigma}_{(t)}^2 = 2\widehat{\text{Var}}\left(\left\{\tilde{\boldsymbol{\theta}}_j^{(t)}\right\}_{j=1}^{M_\epsilon}\right)$$

end for

Output: Samples $\left(\tilde{\boldsymbol{\theta}}_1^{(T)}, \ldots, \tilde{\boldsymbol{\theta}}_{M_\epsilon}^{(T)}\right)$ from the approximate posterior

To improve the posterior approximation and speed up the convergence, we combine the aforementioned two methods by applying the regression adjustment step on the accepted parameters after each iteration of PMC-ABC. The sample mean of the accepted samples after T^{th} iteration, $(\tilde{\boldsymbol{\theta}}_1^{(T)}, \ldots, \tilde{\boldsymbol{\theta}}_{M_\epsilon}^{(T)})$, gives the approximate MMSE estimate.

B. DEEP LEARNING

Given a set of summary statistics and corresponding model parameters, $\{(\mathbf{s}_i, \boldsymbol{\theta}_i)\}_{i\in\{1,\ldots,M\}}$, the calibration problem described above can be expressed as a mapping from $\mathbf{s}$ to $\boldsymbol{\theta}$. Denoting f as the mapping function, the model parameters can be expressed as

$$\boldsymbol{\theta}_i = f(\mathbf{s}_i) + \varepsilon_i \quad i = 1, \ldots, M, \tag{5}$$

where ε_i denotes the approximation error. Given the expression in (5), the calibration problem is equivalent to finding a representation of f such that ε_i is minimized for all $\mathbf{s}_i$. The function f can, e.g., be defined as a linear or polynomial function of $\mathbf{s}$ which can be obtained via a least square fit to the simulated data, $\{\mathbf{s}_i\}_{i=1}^M$. It may however be difficult to find such functions for the multi-dimensional and potentially complex relationship between the statistics and the model parameters. We, therefore, propose using a deep neural network architecture which has been shown to exhibit universal approximation property [26].

The DL based calibration method is illustrated in Fig. 3. We approximate f using a deep NN architecture illustrated in Fig. 4. The deep NN model can be defined using a hypothesis, $\hat{f}(\cdot; \boldsymbol{\Phi})$, with parameter $\boldsymbol{\Phi}$. For a network with R hidden layers, E_r neurons in the the r^{th} hidden layer, and p neurons in the output layer (since $\boldsymbol{\theta} \in \mathbb{R}^p$), the DL hypothesis for the i^{th} statistic can be expressed as [26]

$$\begin{aligned} \mathbf{h}_r(\mathbf{s}_i) &= a_h(\mathbf{W}_r \mathbf{h}_{r-1}(\mathbf{s}_i) + \mathbf{b}_r), \quad r = 1, \ldots, R \\ \hat{f}(\mathbf{s}_i; \boldsymbol{\Phi}) &= a_{\text{out}}(\mathbf{W}_{\text{out}} \mathbf{h}_R(\mathbf{s}_i) + \mathbf{b}_{\text{out}}) \end{aligned} \tag{6}$$

where $\mathbf{h}_r(\cdot)$ is the output of the r^{th} layer, $\mathbf{W}_r \in \mathbb{R}^{E_r \times E_{r-1}}$ and $\mathbf{W}_{\text{out}} \in \mathbb{R}^{p \times E_R}$ denote the weights matrix for connections terminating at the r^{th} hidden layer and output layer, respectively, with $\mathbf{b}_r \in \mathbb{R}^{E_r}$ and $\mathbf{b}_{\text{out}} \in \mathbb{R}^p$ being the corresponding bias terms. The activation function at the nodes of each intermediate layer is a_h, and that at the output layer is a_{out}. Note that $\mathbf{h}_0(\mathbf{s}_i) = \mathbf{s}_i$, with the number of neurons in the input layer being the dimensionality of $\mathbf{s}$. The DL hypothesis in (6) is parameterized by the set $\boldsymbol{\Phi} = \{\{(\mathbf{W}_r, \mathbf{b}_r)\}_{r=1}^R, (\mathbf{W}_{\text{out}}, \mathbf{b}_{\text{out}})\}$. The network parameters are estimated by training the network using the simulated data-set, $\{(\mathbf{s}_i, \boldsymbol{\theta}_i)\}_{i\in\{1,\ldots,M\}}$, from the model. Typically, the training is done by minimizing a loss function, $\mathcal{L}(\boldsymbol{\Phi})$, defined as

$$\mathcal{L}(\boldsymbol{\Phi}) = \frac{1}{2M} \sum_{i=1}^{M} \|f(\boldsymbol{\Phi}, \mathbf{s}_i) - \boldsymbol{\theta}_i\|_2. \tag{7}$$

Minimization of the loss function is typically performed via stochastic gradient descent with back propagation of the error gradients viz:

$$\boldsymbol{\Phi}_n = \boldsymbol{\Phi}_{n-1} - \zeta \nabla \mathcal{L}(\boldsymbol{\Phi}_{n-1}), \tag{8}$$

where ζ denotes the step-size (also referred to as the learning rate) and ∇ is the gradient operator. Due to its fast convergence and good generalization for small data-sets, we utilize the Levenberg – Marquardt algorithm [36], [37] for network training in this paper. The network parameters are thus updated as

$$\boldsymbol{\Phi}_n = \boldsymbol{\Phi}_{n-1} - \left[\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I}\right]^{-1} \mathbf{J}^T \mathcal{L}(\boldsymbol{\Phi}_{n-1}), \tag{9}$$

where $\mathbf{J}$ is the Jacobian matrix with $J_{m\ell} = \delta(f(\boldsymbol{\Phi}, \mathbf{s}_m) - \boldsymbol{\theta}_m)/\delta\Phi_\ell$, $\mathbf{I}$ is an identity matrix and λ is the adaptive damping factor. The damping factor controls the learning rate and is increased by λ_{inc} or decreased by λ_{dec} with increasing or decreasing error, respectively. This procedure is repeated until a termination criterion is achieved. Implementation of

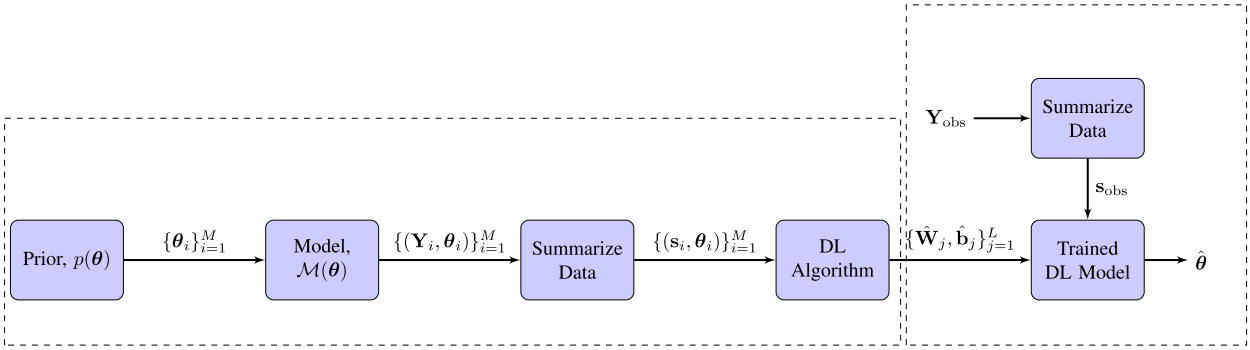

FIGURE 3. Block diagram for the proposed DL method. The left and right dashed boxes represent the training and the test phase, respectively.

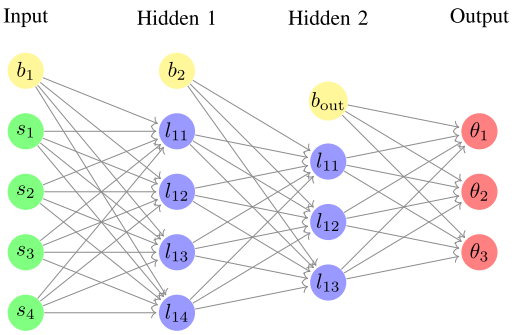

FIGURE 4. Description of a general deep neural network architecture with two hidden layers.

such a NN architecture can be achieved through standard toolboxes available in MATLAB, R, or Python.

Denoting the trained network parameters as $\hat{\boldsymbol{\Phi}}$, the calibration is done by applying the trained network on $\mathbf{s}_{\text{obs}}$ as

$$\hat{\boldsymbol{\theta}} = \hat{f}\left(\mathbf{s}_{\text{obs}}; \hat{\boldsymbol{\Phi}}\right). \tag{10}$$

The accuracy of the estimated model parameter, $\hat{\boldsymbol{\theta}}$, is affected by how well the trained hypothesis approximates the relationship between the summary statistics and the model parameters. This is dependent upon a number of factors such as the selected network structure, activation functions, and training method. Therefore, adequate care has to be taken in selecting the NN model in order to obtain reasonable estimates.

III. CALIBRATION OF POLARIMETRIC PROPAGATION GRAPH MODEL

The proposed ABC and DL methods are applied for estimating parameters of a polarimetric propagation graph (PG) model [30], [38], [39]. First proposed in [40], the PG offers a simple and efficient approach for modeling propagation channels that account for both specular and dense multipath components. The model also has the ability to capture reverberation effects. Several studies have applied or modified the PG model in indoor [41], outdoor-to-indoor [42], high-speed railway [43], [44], indoor-to-indoor [45], [46] and millimetre-wave systems [47], [48]. Despite the growing interest in the PG model, study on its calibration based on measurements is severely limited. A vast majority of works utilizing the PG model are either based on the stochastic generation procedure in [49] or in combination with a map of the environment. The polarimetric PG model in [30] was calibrated with measurements using method of moments. However, the method requires manually fixing one parameter due to identifiability issues. Moreover, the measurement noise variance is not estimated, necessitating manual truncation of the power delay profile (PDP) prior to fitting.

A. MODEL DESCRIPTION

Consider a time-invariant radio channel in a multi-input, multi-output (MIMO) set-up with N_t and N_r output ports at the transmit and receive antennas, respectively. In the PG framework , the radio channel is modeled as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ [49]. The vertex set $\mathcal{V} = \mathcal{V}_t \cup \mathcal{V}_r \cup \mathcal{V}_s$ is a union of a set $\mathcal{V}_t$ of N_t transmitters, a set $\mathcal{V}_r$ of N_r receivers, and a set $\mathcal{V}_s$ of N_s scatterers in the environment. The edges $\mathcal{E} = \mathcal{E}_{\text{d}} \cup \mathcal{E}_{\text{t}} \cup \mathcal{E}_{\text{s}} \cup \mathcal{E}_{\text{r}}$ model the wave propagation between the vertices, where $\mathcal{E}_{\text{d}}$ is a set of direct edges, $\mathcal{E}_{\text{t}}$ is a set of transmitter to scatterer edges, $\mathcal{E}_{\text{s}}$ is a set of scatterer to scatterer edges and $\mathcal{E}_{\text{r}}$ is a set of scatterer to receiver edges.

To each vertex v we associate a position $\mathbf{r}_v \in \mathbb{R}^3$. From these positions, the length of an edge (v, w) is $||\mathbf{r}_v - \mathbf{r}_w||$. This results in a propagation delay from v to w of $\tau_e = \|(\mathbf{r}_w - \mathbf{r}_v)\|/c$, where c is the speed of light in vacuum and $\|\cdot\|$ denotes the Euclidean norm. Accordingly, the direction of propagation is specified by a unit vector Ω_e associated with edge e, pointing in the direction of propagation.

The transfer function matrix at a particular frequency, $\mathbf{H}(f)$, of the polarimetric PG is given as

$$\mathbf{H}(f) = \mathbf{D}(f) + \mathbf{R}(f)\left[\mathbf{I} - \mathbf{B}(f)\right]^{-1}\mathbf{T}(f), \tag{11}$$

where $\mathbf{D}(f) \in \mathbb{C}^{N_{\text{r}} \times N_{\text{t}}}$ is the transmitter to receiver, $\mathbf{T}(f) \in \mathbb{C}^{2N_{\text{s}} \times N_{\text{t}}}$ is the transmitter to scatterer, $\mathbf{R}(f) \in \mathbb{C}^{N_{\text{r}} \times 2N_{\text{s}}}$ is the scatterer to receiver, and $\mathbf{B}(f) \in \mathbb{C}^{2N_{\text{s}} \times 2N_{\text{s}}}$ is the scatterer to scatterer edge transfer function sub-matrix. Then, the transfer function sub-matrices are given as:

$$\mathbf{D}(f) = \mathcal{X}_{\text{t}}^T(\Omega_e)\mathcal{X}_{\text{r}}(\Omega_e)G_e(f), \quad e \in \mathcal{E}_{\text{d}}$$

$$\mathbf{T}(f) = \mathcal{X}_{\mathrm{t}}^{T}(\Omega_e)\mathbf{M}\boldsymbol{\Gamma}(\Omega_e)G_e(f), \quad e \in \mathcal{E}_{\mathrm{t}}$$
$$\mathbf{B}(f) = \mathbf{M}\boldsymbol{\Gamma}(\Omega_e)G_e(f), \quad e \in \mathcal{E}_{\mathrm{s}}$$
$$\mathbf{R}(f) = \mathcal{X}_{\mathrm{r}}(\Omega_e)G_e(f), \quad e \in \mathcal{E}_{\mathrm{r}}$$

Here, $\mathcal{X}_{\mathrm{t}}(\Omega_e)$ and $\mathcal{X}_{\mathrm{r}}(\Omega_e)$ are the 2×1 transmit and receive polarimetric antenna array response vectors, respectively, and $\boldsymbol{\Gamma}(\Omega_e)$ is the 2×2 rotation matrix. The 2×2 scattering matrix, $\mathbf{M}$, represents the coupling between the two polarization states. Assuming it is equal for all the scatterers, $\mathbf{M}$ reads

$$\mathbf{M} = \frac{1}{1+\gamma}\begin{bmatrix} 1 & \gamma \\ \gamma & 1 \end{bmatrix}, \tag{12}$$

where $\gamma \in (0, 1)$ is the polarization power coupling ratio. Finally, $G_e(f)$ is the scalar that captures polarization-independent propagation characteristics, and is expressed as

$$G_e(f) = g_e(f)\exp\big[j(\psi_e - 2\pi\tau_e f)\big], \tag{13}$$

where ψ_e is the phase. The edge gain, $g_e(f)$ is calculated as:

$$g_e(f) = \begin{cases} \frac{1}{(4\pi f\tau_e)}; & e \in \mathcal{E}_{\mathrm{d}} \\ \frac{1}{\sqrt{4\pi\tau_e^2 f\mu(\mathcal{E}_{\mathrm{t}})\mathcal{S}(\mathcal{E}_{\mathrm{t}})}}; & e \in \mathcal{E}_{\mathrm{t}} \\ \frac{g}{\mathrm{odi}(e)}; & e \in \mathcal{E}_{\mathrm{s}} \\ \frac{1}{\sqrt{4\pi\tau_e^2 f\mu(\mathcal{E}_{\mathrm{r}})\mathcal{S}(\mathcal{E}_{\mathrm{r}})}}; & e \in \mathcal{E}_{\mathrm{r}} \end{cases} \tag{14}$$

Here, $g \in (0, 1)$ is the reflection gain, $\mathrm{odi}(e)$ denotes the number of outgoing edges from the n^{th} scatterer, and

$$\mu(\mathcal{E}_{\mathrm{a}}) = \frac{1}{|\mathcal{E}_{\mathrm{a}}|}\sum_{e\subset\mathcal{E}_{\mathrm{a}}}\tau_e, \quad \mathcal{S}(\mathcal{E}_{\mathrm{a}}) = \sum_{e\subset\mathcal{E}_{\mathrm{a}}}\tau_e^{-2}, \quad \mathcal{E}_{\mathrm{a}} \subset \mathcal{E},$$

with $|\cdot|$ denoting set cardinality.

To draw a random graph and simulate transfer function from the model, the positions of the transmit and receive antennas need to be specified. An edge between $\mathcal{V}_{\mathrm{t}}$ and $\mathcal{V}_{\mathrm{r}}$ is drawn with probability P_{dir}. Note that for line-of-sight case, $P_{\mathrm{dir}} = 1$, while for non-line-of-sight (NLOS) case $P_{\mathrm{dir}} = 0$. Edges between $\mathcal{V}_{\mathrm{t}}$ and $\mathcal{V}_{\mathrm{s}}$, $\mathcal{V}_{\mathrm{s}}$ and $\mathcal{V}_{\mathrm{s}}$, or $\mathcal{V}_{\mathrm{s}}$ and $\mathcal{V}_{\mathrm{r}}$ are drawn with probability P_{vis}. The phase ψ_e is drawn uniformly between 0 and 2π.

B. CALIBRATION PROBLEM FORMULATION

To calibrate the PG model based on measured data, $\mathbf{Y}_{\mathrm{obs}}$, we need to estimate the parameters of the model such that the model fits the data. We consider measurements conducted in NLOS conditions, resulting in $P_{\mathrm{dir}} = 0$. Apart from the model parameters, we would also like to estimate the noise variance. The parameter vector to be estimated from $\mathbf{Y}_{\mathrm{obs}}$ thus becomes $\boldsymbol{\theta} = [g, N_s, P_{\mathrm{vis}}, \gamma, \sigma_N^2]^T$.

C. MEASUREMENT DATA DESCRIPTION

Let the MIMO channel transfer function be measured at K equidistant points in the bandwidth B, giving a frequency separation of $\Delta f = B/(K-1)$. The resulting measured signal matrix at each frequency point, $\mathbf{Y}_k \in \mathbb{C}^{N_r\times N_t}$, reads

$$\mathbf{Y}_k = \mathbf{H}(f_k) + \mathcal{N}_k, \quad k = 0, 1, \ldots, K-1 \tag{15}$$

where $\mathbf{H}(f_k)$ is the sampled transfer matrix, and $\mathcal{N}_k$ is the measurement noise. Assuming independent and identically distributed (iid) noise at each measurement point and for each transmitter-receiver link, we model it as iid zero-mean complex Gaussian variables with variance σ_N^2. The entire polarized observed data-set, denoted as $\mathbf{Y}_{\mathrm{obs}}$, thus becomes an $N_r \times N_t \times K$ matrix. Let Y_k^{ij} be the measurement for the k^{th} frequency sample between the i^{th} receiver and the j^{th} transmitter. The received signal in time-domain, $y^{\mathrm{ij}}(t)$, is computed as

$$y^{\mathrm{ij}}(t) = \frac{1}{K}\sum_{k=0}^{K-1} Y_k^{\mathrm{ij}}\exp(j2\pi k\Delta f t). \tag{16}$$

D. SUMMARY STATISTICS

Implementation of the ABC and the DL learning method necessitates a choice of appropriate summary statistics of observed data that are informative about the model parameters. The first three temporal moments, computed as

$$m_l^{\mathrm{ij}} = \int_0^{1/\Delta f} t^l |y^{\mathrm{ij}}(t)|^2, \quad l = 0, 1, 2, \tag{17}$$

have been used previously for calibration of stochastic radio channel models and found to be informative about model parameters [16]–[19]. Here, we compute the sample mean of the l^{th} temporal moment as

$$\bar{m}_l = \frac{1}{N_r N_t}\sum_{i=1}^{N_r}\sum_{j=1}^{N_t} m_l^{\mathrm{ij}}, \tag{18}$$

and the sample covariance between l^{th} and l'^{th} temporal moment as

$$\widehat{\mathrm{cov}}\Big(m_l^{\mathrm{ij}}, m_{l'}^{\mathrm{ij}}\Big) = \frac{1}{N_r N_t - 1}\sum_{i=1}^{N_r}\sum_{j=1}^{N_t}\Big(m_l^{\mathrm{ij}} - \bar{m}_l\Big)\Big(m_{l'}^{\mathrm{ij}} - \bar{m}_{l'}\Big). \tag{19}$$

Additionally, we separate the temporal moments according to their polarization, i.e., vertical-vertical (vv), vertical-horizontal (vh), horizontal-vertical (hv), and horizontal-horizontal (hh), to compute the cross-polarization ratio, XPR, as

$$\mathrm{XPR} = \frac{1}{2}\left[\frac{\bar{m}_0^{vv}}{\bar{m}_0^{vh}} + \frac{\bar{m}_0^{hh}}{\bar{m}_0^{hv}}\right]. \tag{20}$$

The summary statistics vector, $\mathbf{s}$, therefore has ten entries: the XPR, the three means $\bar{m}_l$, and, and six covariances $\widehat{\mathrm{cov}}(m_l^{\mathrm{ij}}, m_{l'}^{\mathrm{ij}})$ for $l, l' = 0, 1, 2$.

To verify that the chosen summary statistics are informative about the model parameters, we conduct a simulation experiment. One parameter at a time is sampled 100 times from its uniform prior distribution (given in Tab. 1), while the other parameters are held fixed. Data is simulated from the polarimetric PG model for such a parameter vector and

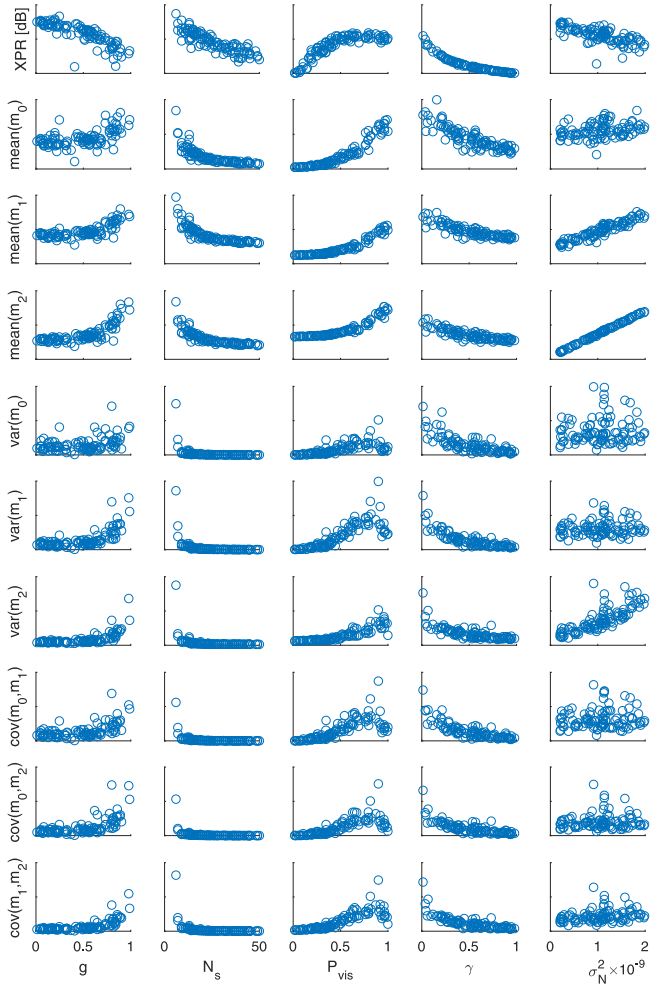

FIGURE 5. Summary statistics versus model parameters. Each plot is generated by varying one parameter while the others are held fixed to the values in Tab. 1.

the statistics are computed. Each of the ten statistics are then plotted versus the five parameters in Fig. 5.

We observe that XPR and the means of the temporal moments are informative about almost all the parameters. The covariances become informative for higher values of g and P_{vis} and lower values of γ. We see a clear functional relationship devoid of any jitter between XPR and γ, and $\bar{m}_2$ and σ_N^2. This indicates from the outset that γ and σ_N^2 should be estimated very accurately. In contrast, the summaries seem the least informative about g, suggesting that the estimate of g would be the most uncertain. In principle, a subset of these statistics could also be used in the calibration methods. However, we observed a degradation in performance on leaving out the covariances, and therefore include all ten statistics.

IV. PERFORMANCE EVALUATION

We apply the ABC and DL methods to calibrate the polarimetric PG model using the summary statistics described in the previous section. First, we evaluate the performance of the two calibration methods via simulations, and later validate it using millimetre-wave NLOS measurements from [50]. The measurements were taken in a room of dimensions $3 \times 4 \times 3$ m^3 in the bandwidth range of 58 GHz to 62 GHz, sampled at $K = 801$ equidistant points. The frequency separation of $\Delta f = 5$ MHz results in a signal observation interval of 200 ns in the time domain. A 5×5 virtual planar array of dual polarized antennas with 5 mm inter-element spacing, was used at both the receiver and the transmitter. This gives $N_t = N_r = 50$.

For the simulation experiment, we set the parameters of the model to some "true" value, say $\boldsymbol{\theta}_{\text{true}}$, and generate data from the model that we consider as observed data. We then apply the proposed methods on this simulated data to estimate the parameters. We use the same settings for simulations as in the measurements [50]. The antennas in the PG model implementation are assumed to be omni-directional with perfect cross-polar isolation at both the receiver and the transmitter. The scatterers are distributed uniformly across the floor of the room.

For the simulation experiment, the observed statistics, $\mathbf{s}_{\text{obs}}$, that corresponds to $\boldsymbol{\theta}_{\text{true}}$ needs to be set. For a fixed $\boldsymbol{\theta}_{\text{true}}$, the stochastic model generates samples $(\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(Z)})$ from $p(\mathbf{s}|\boldsymbol{\theta}_{\text{true}})$. Running the estimator Z times with each realization of the statistics vector as $\mathbf{s}_{\text{obs}}$ results in Z parameter estimates, giving the error distribution around $\boldsymbol{\theta}_{\text{true}}$. However, as shown in the Appendix, this is equivalent to taking $\mathbf{s}_{\text{obs}}$ as the sample mean of $(\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(Z)})$ and running the estimator once. Here, we adopt the latter computationally convenient approach and compute $\mathbf{s}_{\text{obs}}$ from $\boldsymbol{\theta}_{\text{true}}$ using $Z = 200$. It should be noted that a similar approach is only possible in general for measured data if Z independent measurements can be obtained.

A. APPROXIMATE BAYESIAN COMPUTATION

The ABC method is applied with $M = 2000$ samples simulated for each iteration. For the first iteration, the samples are taken from a uniform prior distribution of the parameters with ranges given in Tab. 1. Note that the prior for N_s is uniform integers in the specified range. A Gaussian kernel truncated to the prior range is used to generate populations of subsequent iterations. We set the tolerance threshold to $\epsilon = 5\%$, giving $M_\epsilon = 100$ accepted samples, and the total number of iterations to $T = 10$. The summary statistics are normalized by the estimate of their median absolute deviation before applying ABC. A logit transformation is applied to the parameters before regression adjustment to keep the adjusted samples within the prior boundary. The estimated marginal posterior distributions are shown in Fig. 6 and Fig. 7 for simulated and measured data, respectively. The obtained point estimates and the sample standard deviation of the accepted samples after $T = 10$ iterations are reported in Tab. 1.

We observe in Fig. 6 that the samples obtained from approximate posteriors lie around the true value for all the parameters, and that the MMSE estimates are fairly accurate

TABLE 1. Summary of the parameter estimates obtained from ABC and DL for simulated and measured data. Note that the sample standard deviation reported in parenthesis is of the approximate posterior for ABC and of the estimate distribution for DL.

Parameter θ	Prior range $p(\theta)$	Estimate (standard deviation)					
		Simulated Data			Measured Data		
		True value	ABC	DL	ABC	DL	MoM [30]
Reflection gain, g	$[0,1]$	0.6	0.59 (0.11)	0.58 (0.04)	0.54 (0.03)	0.56 (0.05)	0.64
Number of scatterers, N_s	$[5,50]$	15	15 (1.56)	15 (0.92)	14 (1.25)	16 (6.5)	11
Probability of visibility, P_{vis}	$[0,1]$	0.6	0.58 (0.04)	0.60 (0.014)	0.99 (0.006)	0.96 (0.03)	0.9[a]
Polarization ratio, γ	$[0,1]$	0.5	0.51 (0.02)	0.50 (0.01)	0.09 (0.005)	0.09 (0.05)	0.06
Noise variance, σ_N^2	$[2\times10^{-10}, 2\times10^{-9}]$	10^{-9}	9.98×10^{-10} (2.86×10^{-12})	9.96×10^{-10} (8.8×10^{-12})	4.3×10^{-10} (2.11×10^{-11})	4.3×10^{-10} (2.8×10^{-11})	-

[a] Note that this value is not estimated but set in [30].

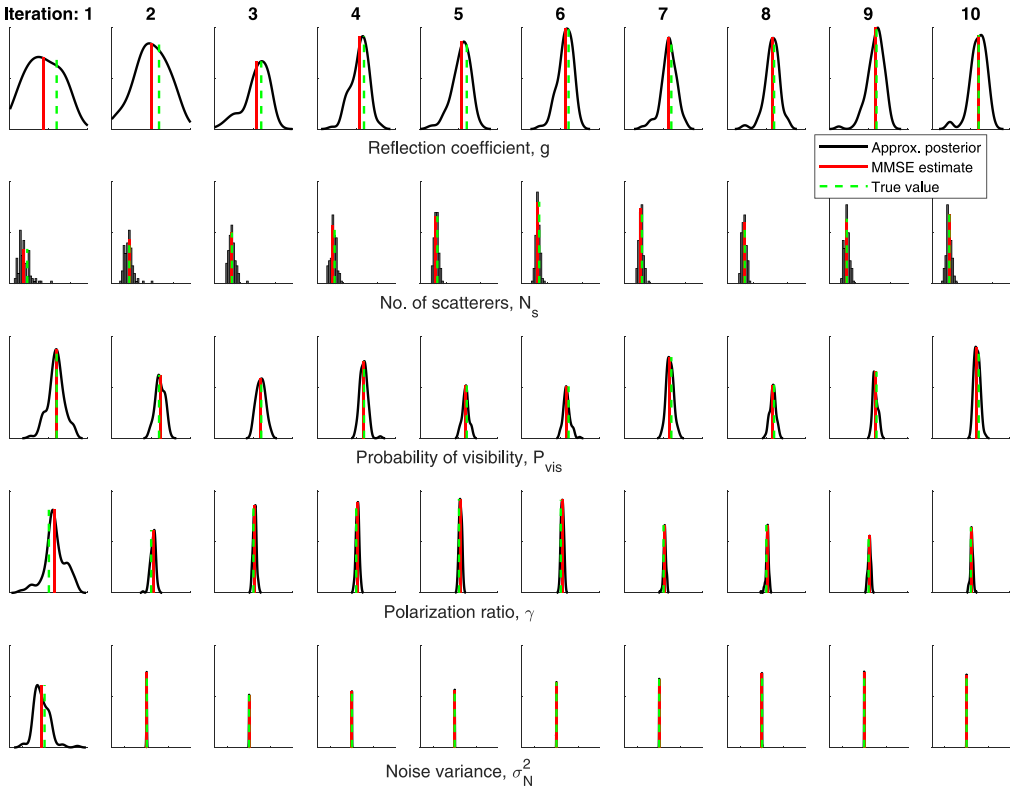

FIGURE 6. Kernel density estimates of the marginal approximate posteriors of the parameters obtained by applying ABC on simulated data, plotted in the prior range for each parameter. The posteriors are shown after each iteration of the algorithm, with the parameter estimate marked in red. The true value of the parameter is shown in green.

even after the first iteration. As the iterations go on, the posteriors shrink and converge for each parameter, albeit some faster than others. For example, the posteriors for N_s, P_{vis}, γ and σ_N^2 barely change after the second iteration, while that of g seem to converge after around five iterations with the MMSE estimate getting better with further iterations. Similarly, the posterior for g is the widest, while that of the other parameters are quite narrow. This uncertainty in the estimates of different parameters reflects their relationship with the summary statistics shown in Fig. 5. Parameters that have a distinct functional relationship with, at least, a few statistics are easier to estimate than others.

Similar behavior is observed in Fig. 7 for measurements as was seen for the simulation experiment. The approximate posteriors for P_{vis}, γ, and σ_N^2 are very narrow, and seem to have converged since the first iteration. However, the posteriors for g and N_s takes approximately four iterations to converge. The width of posterior for g and P_{vis} is narrower in comparison to those in simulation. This is attributed to the fact that for a high value of P_{vis}, as is the case in measurements, almost all the statistics become informative, see Fig. 5. The averaged power delay profile (APDP) generated from the model using the point estimates obtained for the measurements is shown in Fig. 11. The calibrated model fits both the co- and cross-polarized APDP of the measurements well.

B. DEEP LEARNING

We determine the structure of the NN for the calibration problem via a guided search procedure. First, we limit the number of hidden layers to $R = 2$ and assume that there are equal number of neurons in each layer, i.e., $E_r = E$.

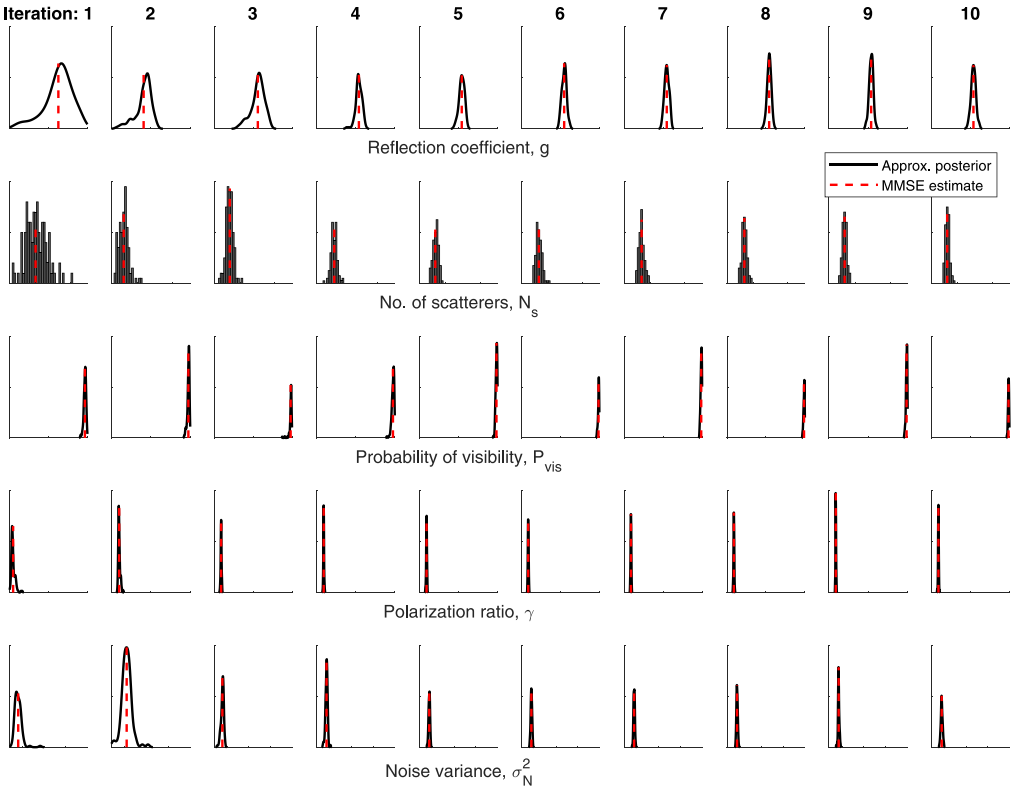

FIGURE 7. Kernel density estimates of the marginal approximate posteriors of the parameters obtained by applying ABC on measured data. The posteriors are shown after each iteration of the algorithm. The density is plotted in the prior range for each parameter. The sample mean is marked in red.

A single hidden layer architecture is excluded due to its poor performance during our preliminary experiments. The number of neurons in each layer is varied from 2 to 28. We then divide the data set into two equal subsets for training and cross-validation. The mean and standard deviation of squared error on the training and validation subsets are shown in Fig. 8. We observe that the error stabilizes after around 12 neurons, and so we set $E = 20$ as this is sufficient. This results in a 10-20-20-5 network architecture which is used for evaluating the DL calibration method. We use the hyperbolic tangent sigmoid and linear activation functions [26] for the hidden and output layers, respectively.

The NN is trained using the same $M = 2000$ samples of $\mathbf{s}_i$ and corresponding $\boldsymbol{\theta}_i$ as used in the initialization of the ABC method. In principle, the training data could be as extensive as possible, thus leading to a better approximation of the summary-parameter function. In order to eliminate the sensitivity of network to the range of values in the summary statistics, the entire data-set is normalized using the standard Z-score scaling prior to network training. The data-set is randomly partitioned into training, test, and validation subsets in the ratio 0.70, 0.15 and 0.15, respectively. We utilized the LM algorithm with damping parameters: $\lambda = 0.1$, $\lambda_{\text{inc}} = 10$ and $\lambda_{\text{dec}} = 0.1$ for training the NN. The training procedure is terminated when the number of epochs reaches 1000 or the gradient is below 10^{-7}.

Once the termination criterion is achieved, we apply the trained network on $\mathbf{s}_{\text{obs}}$ from simulated and measured data to get point estimate of the parameter vector. This process is repeated 200 times to estimate the distribution of the parameter estimates, which is shown in Fig. 9 and Fig. 10 for simulated and measured data, respectively. The sample mean of the estimates and their standard deviations are reported in Tab. 1. We observe in Fig. 9 that the DL method is able to estimate the model parameters accurately and with reasonable precision. The uncertainty in the estimates is fairly small. As was the case with the ABC method, the estimate of g has the largest standard deviation out of all the parameters, corroborating our conjecture based on Fig. 5. The method performs similarly on measured data, as seen in Fig. 10, although with slightly larger standard deviations. The APDP generated from the parameter estimates fits the measurements, see Fig. 11, thus validating the methodology.

V. DISCUSSION

In our example, we see that the calibration approach based on summaries is effective. However, as experienced during the development of the algorithms, the choice of summaries is important for obtaining a good calibration. Ideally, sufficient statistics should be considered, but such are rarely available or practical to extract in the context of radio channel models. Although the simulation method of checking summaries has proved useful for the example problem considered here, it only gives an indication of how informative the summaries are, and does not guarantee that the algorithms will work.

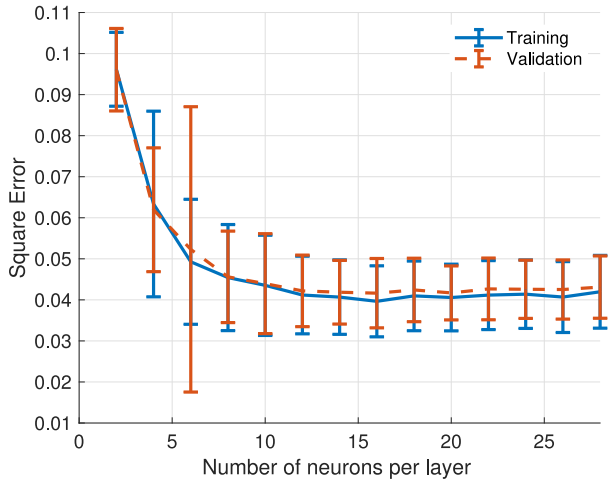

FIGURE 8. Learning curve for selecting NN size. The mean and standard deviation of the squared error at each number of neurons per layer is computed from 200 repeated network training.

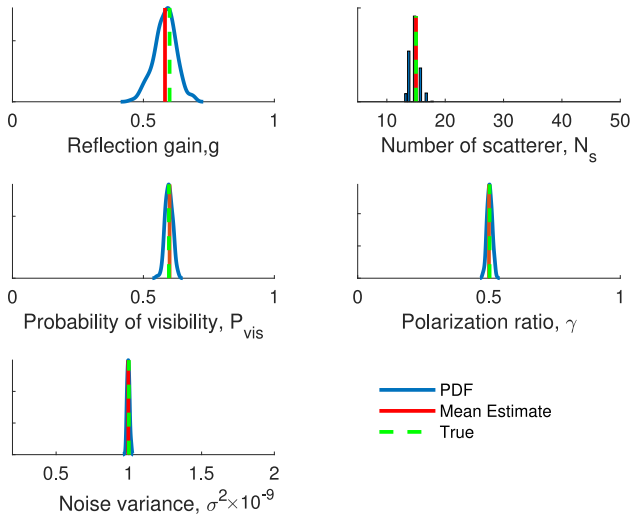

FIGURE 9. Parameter estimates obtained by applying the DL method on simulated data. Kernel density estimates of the distribution obtained after 200 estimator runs.

The summary statistics used in this paper appear to be informative about the different aspects of the channel, and could be useful in calibrating other stochastic channel models as well. In the case of calibrating a directional model, the summaries possibly have to be chosen differently. The method for checking the summaries described here would be useful for this selection.

Although based on the summary statistics of the data, the two proposed methods are complementary to each other as they approach the same problem in distinct ways. To highlight this fact, we have intentionally avoided comparing the two methods with each other. ABC infers on the parameters in a Bayesian sense by learning the distribution of the parameters given the data. On the other hand, the DL method fits a function between the summaries and the parameters and provides point estimates of the parameters given the data in a frequentist manner without considering any prior information. Despite their differences, both methods are able to learn the parameters of the model such that, qualitatively, it is not apparent in Fig. 11 which APDP fits the measurements better.

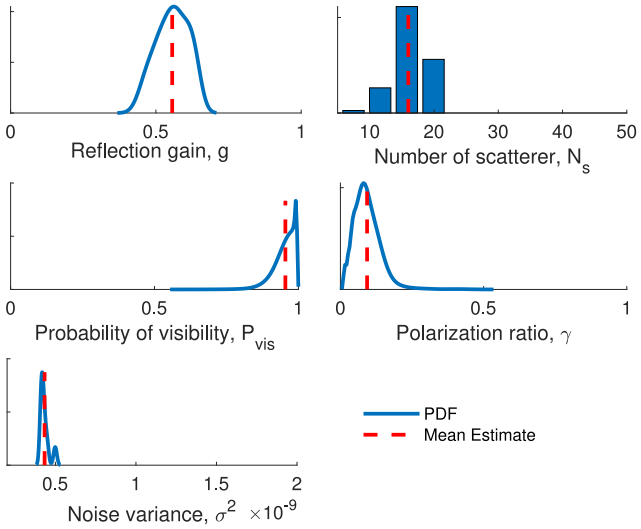

FIGURE 10. Parameter estimates obtained by applying the DL method on measured data. Kernel density estimates of the distribution obtained after 200 estimator runs.

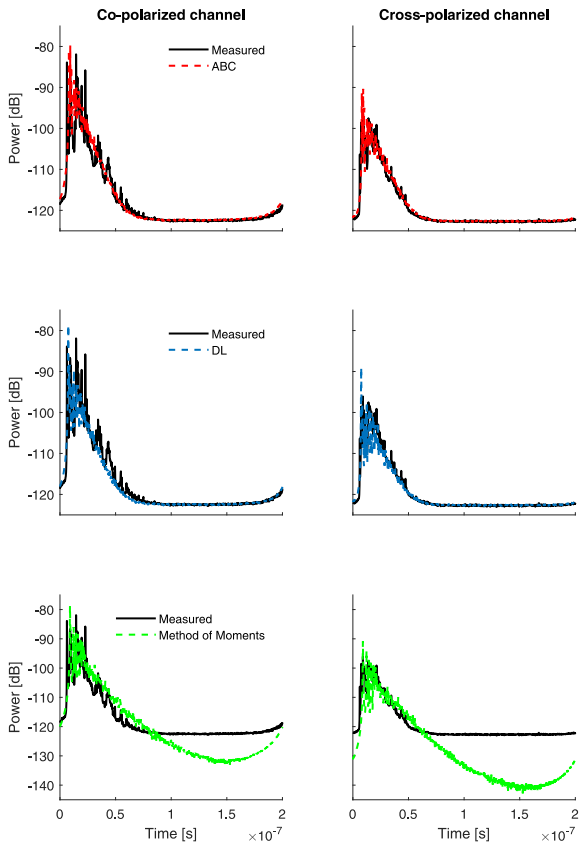

FIGURE 11. Averaged power delay profile from the measurements versus those obtained from the polarimetric PG model after calibration using ABC, DL and the method of moments approach from [30]. The parameter estimates for all three methods are reported in Tab. 1.

The choice of prior naturally affects the posterior distribution of the ABC method. In this paper, we used uninformative (flat) priors in order to rely solely on the data to estimate the parameters. The ranges of these priors are chosen conservatively, i.e., to be very wide. Choosing more informative priors instead would lead to faster convergence of the approximate posterior in the ABC method, thus reducing run-time. In general, if the data is large enough, the effect of the prior distribution on the posterior becomes irrelevant.

Since the ABC method relies on simulations from the model, the computational complexity primarily depends on the complexity of the model. The computationally expensive step in the DL method is the training of the NN, which depends on the size of the training data and the chosen NN architecture. While the two layer architecture was found to be sufficient for the example model in the paper, more complex models may require deeper networks. However, the network needs to be trained just once, and parameter estimates can be obtained for different measurement data instantly. In contrast, the sequential nature of the ABC algorithm requires running the iterations again for new observed statistics.

The computation time for the proposed methods depends on the particular implementation and the available hardware. The methods are lightweight enough to be run on standard laptops with reasonable run-time. As an indication of the required run-time of the proposed methods, our implementation of the proposed ABC method was able to complete ten iterations within a day on a Lenovo ThinkPad with Intel Core i7 processor having 24 GB RAM. Training the NN took less than 2 hours on a Lenovo ThinkPad with Intel Core i7 processor and 16 GB RAM. Due to the different choices of high-resolution and clustering algorithms that are available, their comparison with the proposed methods in terms of computational complexity is not feasible. We remark that in all the cases, the specific run-time depends not only on the choice of algorithms and hardware, but is significantly impacted by the particular implementation and choice of settings in the respective algorithms.

We experience that the ABC method is not very sensitive to the particular settings of the algorithm due to its iterative nature. Increasing the number of simulated samples per iteration, M, increases simulation time from the model, but leads to lower ϵ if M_ϵ is kept constant. Thus, the algorithm would converge in fewer iterations. Similarly, increasing ϵ would mean accepting samples that are further away from $\mathbf{s}_{\text{obs}}$, and therefore, would require more iterations to converge to a stable approximate posterior. Overall, changes in one setting is compensated by another, and the method performs similarly. This means that the performance primarily relies on the choice of statistics. In contrast, for the DL method the particular NN architecture should be chosen carefully since generalization accuracy is sensitive to the network size, particularly with small number of training samples.

In principle, the entire channel impulse response measurements could be used as input data instead of a set of summary statistics. However, the ABC method is then hit by the curse of dimensionality [35]. That is, the distances become very large due to the high-dimensional data, thus increasing the rejection rate significantly. For the DL method, this would increase the number of input layers, thereby increasing the computational complexity, along with complicating the training procedure.

An appealing feature of the summary-based approach is that it is specified exactly which criteria are used to fit the model to the data. In this sense, we obtain the best fitting model in the eyes of the summary statistics. This is very different from what is obtained by the multi-step methods relying on ad hocery and possibly conflicting assumptions in the individual steps, e.g., assuming "well-separated" paths in multipath extraction, followed by application of clustering algorithms. Statistical techniques to construct a more informative subset of summaries [32] to be used in ABC can be explored. However, by doing so we lose the transparency as to which summaries are being fitted. Both the proposed machine learning methods are also simpler to implement than the classical state-of-the-art approach of multipath extraction, with fewer settings of the algorithm. Moreover, since the proposed methods are integrated, their performance is easy to investigate by simulation studies. This is again a great advantage compared to multi-step approaches where the performance of each step is evaluated separately, thus making it difficult to judge the accuracy of the overall parameter estimates.

Additionally, the ABC and DL methods are able to estimate all model parameters, including the noise variance. Thus it is not necessary to provide side information to the algorithm (as is often done by separately estimating, e.g., noise variance) or to post-process the data by setting the noise threshold. This advantage is clearly seen when comparing with the method of moments (MoM) approach of calibrating polarimetric PG models [30] where noise is not estimated (see Fig. 11).

VI. CONCLUSION

The proposed machine learning methods based on ABC and deep learning are able to accurately calibrate stochastic radio channel models. The model fit is obtained in light of the explicitly chosen summaries. The proposed methods demonstrate that stochastic channel models, in particular the PG model, can be calibrated without access to likelihoods. The methods also by-pass any intermediate step of extracting the multipath components. We observed that the choice of summaries is crucial in learning the parameters, and the uncertainty in parameter estimates decreases with informative summaries. The summaries used in this paper are general purpose, and we conjecture that they can work for other models as well. The methodology to qualify the summaries through simulation study is useful in the design of the algorithm, although it does not provide any guarantees. Availability of pseudocodes and libraries make the proposed methods easy to implement, compared to the state-of-the-art

approach. The performance of the proposed methods is easy to evaluate, as opposed to the multi-step approach. Moreover, no additional information or post-processing is required to calibrate the model.

APPENDIX

We want to show that setting $\mathbf{s}_{\text{obs}}$ as one realization out of $p(\mathbf{s}|\boldsymbol{\theta}_{\text{true}})$ and running the estimator Z times gives the same estimate of $\boldsymbol{\theta}_{\text{true}}$ in mean as taking $\mathbf{s}_{\text{obs}}$ as the sample mean of Z such realizations out of $p(\mathbf{s}|\boldsymbol{\theta}_{\text{true}})$ and running the estimator once.

Let $\mathbf{s}_{\text{obs}}$ be the z^{th} sample, $\mathbf{s}^{(z)}$, from the distribution $p(\mathbf{s}|\boldsymbol{\theta}_{\text{true}})$, and the corresponding MMSE estimate be $\hat{\boldsymbol{\theta}}^{(z)}$. Then, the sample mean for Z such estimates is

$$\hat{\boldsymbol{\theta}}_{\text{avg}} = \frac{1}{Z}\sum_{z=1}^{Z}\hat{\boldsymbol{\theta}}^{(z)} \approx \mathbb{E}[\boldsymbol{\theta}|\mathbf{s} = \mathbb{E}[\mathbf{s}|\boldsymbol{\theta}_{\text{true}}]]. \tag{21}$$

The mean of $\hat{\boldsymbol{\theta}}^{(z)}$ then reads

$$\mathbb{E}\left[\hat{\boldsymbol{\theta}}^{(z)}\right] = \int \hat{\boldsymbol{\theta}}^{(z)} p(\mathbf{s}|\boldsymbol{\theta}_{\text{true}})\mathrm{d}\mathbf{s} = \boldsymbol{\theta}_{\text{true}}. \tag{22}$$

Assuming $(\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(Z)})$ are independent samples, the expected value of $\hat{\boldsymbol{\theta}}_{\text{avg}}$ can be computed as:

$$\begin{aligned}
&\mathbb{E}\left[\hat{\boldsymbol{\theta}}_{\text{avg}}\right] \\
&= \int \cdots \int \frac{1}{Z}\sum_{z=1}^{Z}\hat{\boldsymbol{\theta}}^{(z)}\Pi_{z'=1}^{Z} p\left(\mathbf{s}^{(z')}|\boldsymbol{\theta}_{\text{true}}\right)\mathrm{d}\mathbf{s}^{(1)}\ldots\mathrm{d}\mathbf{s}^{(Z)} \\
&= \frac{1}{Z}\sum_{z=1}^{Z}\int \hat{\boldsymbol{\theta}}^{(z)} p\left(\mathbf{s}^{(z)}|\boldsymbol{\theta}_{\text{true}}\right)\mathrm{d}\mathbf{s}^{(z)} \\
&= \frac{1}{Z}\sum_{z=1}^{Z}\boldsymbol{\theta}_{\text{true}} = \boldsymbol{\theta}_{\text{true}}
\end{aligned}$$

Therefore, both $\hat{\boldsymbol{\theta}}^{(z)}$ and $\hat{\boldsymbol{\theta}}_{\text{avg}}$ converge to $\boldsymbol{\theta}_{\text{true}}$ in mean and thus, are unbiased estimates. The variance of $\hat{\boldsymbol{\theta}}_{\text{avg}}$ is, however, reduced by a factor of $1/Z$.

ACKNOWLEDGMENT

The authors would like to thank Dr. Carl Gustafson and Prof. Fredrik Tufvesson (Lund University) for providing the measurement data.

REFERENCES

[1] L. Greenstein, S. S. Ghassemzadeh, S.-C. Hong, and V. Tarokh, "Comparison study of UWB indoor channel models," *IEEE Trans. Wireless Commun.*, vol. 6, no. 1, pp. 128–135, Jan. 2007.

[2] G. L. Turin, F. D. Clapp, T. L. Johnston, S. B. Fine, and D. Lavry, "A statistical model of urban multipath propagation," *IEEE Trans. Veh. Technol.*, vol. 21, no. 1, pp. 1–9, Feb. 1972.

[3] A. A. M. Saleh and R. Valenzuela, "A statistical model for indoor multipath propagation," *IEEE J. Sel. Areas Commun.*, vol. 5, no. 2, pp. 128–137, Feb. 1987.

[4] K. Haneda, J. Järveläinen, A. Karttunen, M. Kyrö, and J. Putkonen, "A statistical spatio-temporal radio channel model for large indoor environments at 60 and 70 GHz," *IEEE Trans. Antennas Propag.*, vol. 63, no. 6, pp. 2694–2704, Jun. 2015.

[5] C. Gustafson, K. Haneda, S. Wyne, and F. Tufvesson, "On mm-wave multipath clustering and channel modeling," *IEEE Trans. Antennas Propag.*, vol. 62, no. 3, pp. 1445–1455, Mar. 2014.

[6] J. Poutanen, K. Haneda, L. Liu, C. Oestges, F. Tufvesson, and P. Vainikainen, "Parameterization of the COST 2100 MIMO channel model in indoor scenarios," in *Proc. Eur. Conf. Antennas Propag. (EUCAP)*, 2011, pp. 3606–3610.

[7] P. Kyösti *et al.*, *WINNER II Channel Models, Part I: Channel Models, Deliverable D1.1.2 V1.2*, document IST-4-027756, WINNER II, 2007.

[8] L. Raschkowski, P. Kyösti, K. Kusume, and E. T. Jämsä, *METIS Channel Models, Deliverable D1.4 V3*, document ICT-317669-METIS/D1.4, METIS, 2015.

[9] B. H. Fleury, M. Tschudin, R. Heddergott, D. Dahlhaus, and K. I. Pedersen, "Channel parameter estimation in mobile radio environments using the SAGE algorithm," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 3, pp. 434–450, Mar. 1999.

[10] R. Thomä, M. Landmann, and A. Richter, "Rimax—A maximum likelihood framework for parameter estimation in multidimensional channel sounding," in *Proc. Int. Symp. Antennas Propag. (ISAP)*, Sendai, Japan, 2004, pp. 53–56.

[11] X. Yin, T. Pedersen, N. Czink, and B. H. Fleury, "Parametric characterization and estimation of bi-azimuth and delay dispersion of individual path components," in *Proc. 1st Eur. Conf. Antennas Propag.*, Nov. 2006, pp. 1–8.

[12] X. Yin *et al.*, "Tracking of time-variant radio propagation paths using particle filtering," in *Proc. IEEE Int. Conf. Commun.*, 2008, pp. 920–924.

[13] G. Steinbock, T. Pedersen, X. Yin, and B. H. Fleury, "Experimental characteristics of indoor wideband MIMO radio channels and their impact on stochastic modelling," in *Proc. IEEE 13th Dig. Signal Process. Workshop 5th IEEE Signal Process. Edu. Workshop*, Jan. 2009, pp. 302–307.

[14] A. Karttunen, C. Gustafson, A. F. Molisch, J. Järveläinen, and K. Haneda, "Censored multipath component cross-polarization ratio modeling," *IEEE Wireless Commun. Lett.*, vol. 6, no. 1, pp. 82–85, Feb. 2017.

[15] W.-D. Wu, C.-H. Wang, C.-C. Chao, and K. Witrisal, "On parameter estimation for ultra-wideband channels with clustering phenomenon," in *Proc. IEEE 68th Veh. Technol. Conf.*, Sep. 2008, pp. 1–5.

[16] A. Bharti, R. Adeogun, and T. Pedersen, "Parameter estimation for stochastic channel models using temporal moments," in *Proc. IEEE Int. Symp. Antennas Propag. USNC-URSI Radio Sci. Meeting*, 2019, pp. 1267–1268.

[17] A. Bharti, R. Adeogun, and T. Pedersen, "Estimator for stochastic channel model without multipath extraction using temporal moments," in *Proc. 20th IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2019, pp. 1–5.

[18] A. Bharti and T. Pedersen, "Calibration of stochastic channel models using approximate Bayesian computation," in *Proc. IEEE GLOBECOM Workshops*, 2019, pp. 1–6.

[19] R. Adeogun, "Calibration of stochastic radio propagation models using machine learning," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 12, pp. 2538–2542, Dec. 2019.

[20] M. A. Beaumont, "Approximate Bayesian computation," *Annu. Rev. Statist. Appl.*, vol. 6, no. 1, pp. 379–403, 2019.

[21] M. A. Beaumont, "Approximate Bayesian computation in evolution and ecology," *Annu. Rev. Ecol. Evol. Syst.*, vol. 41, pp. 379–406, Dec. 2010.

[22] J. K. Pritchard, M. T. Seielstad, A. Perez-Lezaun, and M. W. Feldman, "Population growth of human Y chromosomes: A study of Y chromosome microsatellites," *Mol. Biol. Evol.*, vol. 16, no. 12, pp. 1791–1798, Dec. 1999.

[23] O. François, M. G. B. Blum, M. Jakobsson, and N. A. Rosenberg, "Demographic history of european populations of *Arabidopsis Thaliana*," *PLoS Genet.*, vol. 4, no. 5, May 2008, Art. no. e1000075.

[24] J. Akeret, A. Refregier, A. Amara, S. Seehars, and C. Hasner, "Approximate Bayesian computation for forward modeling in cosmology," *J. Cosmol. Astroparticle Phys.*, vol. 2015, no. 8, p. 043, Aug. 2015. [Online]. Available: https://www.research-collection.ethz.ch/handle/20.500.11850/104029

[25] A. B. Abdessalem, N. Dervilis, D. Wagg, and K. Worden, "Model selection and parameter estimation in structural dynamics using approximate Bayesian computation," *Mech. Syst. Signal Process.*, vol. 99, pp. 306–325, Jan. 2018.
[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[27] C. Huang, G. C. Alexandropoulos, A. Zappone, C. Yuen, and M. Debbah, "Deep learning for UL/DL channel calibration in generic massive MIMO systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.
[28] J.-Y. Lee, M. Y. Kang, and S.-C. Kim, "Path loss exponent prediction for outdoor millimeter wave channels through deep learning," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–5.
[29] S. Sheehan and Y. S. Song, "Deep learning for population genetic inference," *PLOS Comput. Biol.*, vol. 12, no. 3, pp. 1–28, Mar. 2016.
[30] R. Adeogun, T. Pedersen, C. Gustafson, and F. Tufvesson, "Polarimetric wireless indoor channel modeling based on propagation graph," *IEEE Trans. Antennas Propag.*, vol. 67, no. 10, pp. 6585–6595, Oct. 2019.
[31] M. A. Beaumont, W. Zhang, and D. J. Balding, "Approximate Bayesian computation in population genetics," *Genetics*, vol. 162, no. 4, pp. 2025–2035, 2002.
[32] M. G. B. Blum, M. A. Nunes, D. Prangle, and S. A. Sisson, "A comparative review of dimension reduction methods in approximate Bayesian computation," *Stat. Sci.*, vol. 28, no. 2, pp. 189–208, May 2013.
[33] D. Prangle, "Adapting the ABC distance function," *Bayesian Anal.*, vol. 12, pp. 289–309, Mar. 2017.
[34] M. A. Beaumont, J.-M. Cornuet, J.-M. Marin, and C. P. Robert, "Adaptive approximate Bayesian computation," *Biometrika*, vol. 96, no. 4, pp. 983–990, Oct. 2009.
[35] M. G. B. Blum, "Approximate Bayesian computation: A nonparametric perspective," *J. Amer. Stat. Assoc.*, vol. 105, no. 491, pp. 1178–1187, Sep. 2010.
[36] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly Appl. Math.*, vol. 2, no. 2, pp. 164–168, 1944.
[37] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994.
[38] R. Adeogun and T. Pedersen, "Propagation graph based model for multipolarized wireless channels," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.
[39] R. Adeogun and T. Pedersen, "Modelling polarimetric power delay spectrum for indoor wireless channels via propagation graph formalism," in *Proc. 2nd URSI Atlantic Radio Sci. Meeting*, May 2018, pp. 1–3.
[40] T. Pedersen and B. H. Fleury, "Radio channel modelling using stochastic propagation graphs," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2007, pp. 2733–2738.
[41] G. Steinböck *et al.*, "Hybrid model for reverberant indoor radio channels using rays and graphs," *IEEE Trans. Antennas Propag.*, vol. 64, no. 9, pp. 4036–4048, Sep. 2016.
[42] T. Pedersen, G. Steinböck, and B. H. Fleury, "Modeling of outdoor-to-indoor radio channels via propagation graphs," in *Proc. URSI Gen. Assembly Sci. Symp.*, Aug. 2014, pp. 1–4.
[43] L. Tian, X. Yin, Q. Zuo, J. Zhou, Z. Zhong, and S. X. Lu, "Channel modeling based on random propagation graphs for high speed railway scenarios," in *Proc. IEEE 23rd Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2012, pp. 1746–1750.
[44] T. Zhou, C. Tao, S. Salous, Z. Tan, L. Liu, and L. Tian, "Graph-based stochastic model for high-speed railway cutting scenarios," *IET Microw. Antennas Propag.*, vol. 9, no. 15, pp. 1691–1697, 2015.
[45] R. O. Adeogun, A. Bharti, and T. Pedersen, "An iterative transfer matrix computation method for propagation graphs in multiroom environments," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 4, pp. 616–620, Apr. 2019.
[46] R. Adeogun, T. Pedersen, and A. Bharti, "Transfer function computation for complex indoor channels using propagation graphs," in *Proc. IEEE Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2018, pp. 566–567.
[47] J. Chen, X. Yin, L. Tian, and M.-D. Kim, "Millimeter-wave channel modeling based on a unified propagation graph theory," *IEEE Commun. Lett.*, vol. 21, no. 2, pp. 246–249, Feb. 2017.
[48] L. Tian, V. Degli-Esposti, E. M. Vitucci, and X. Yin, "Semi-deterministic radio channel modeling based on graph theory and ray-tracing," *IEEE Trans. Antennas Propag.*, vol. 64, no. 6, pp. 2475–2486, Jun. 2016.
[49] T. Pedersen, G. Steinböck, and B. H. Fleury, "Modeling of reverberant radio channels using propagation graphs," *IEEE Trans. Antennas Propag.*, vol. 60, no. 12, pp. 5978–5988, Dec. 2012.
[50] C. Gustafson, D. Bolin, and F. Tufvesson, "Modeling the polarimetric mm-wave propagation channel using censored measurements," in *Proc. Global Commun. Conf.*, Dec. 2016, pp. 1–6.

AYUSH BHARTI (Graduate Student Member, IEEE) received the B.E. degree in electrical and electronics engineering from the Birla Institute of Technology and Sciences, Pilani, India, in 2015, and the M.Sc. degree in signal processing and computing from Aalborg University, Denmark, in 2017, and where he is currently pursuing the Ph.D. degree with the Department of Electronic Systems. His research interests include statistical signal processing, radio channel modeling, and Bayesian inference.

RAMONI ADEOGUN (Senior Member, IEEE) received the B.Eng. degree in electrical and computer engineering from the Federal University of Technology, Minna, Nigeria, and the Ph.D. degree in electronic and computer systems engineering from the Victoria University of Wellington, New Zealand. He has also worked in various positions for the University of Cape Town, Cape Town, South Africa, Odua Telecoms Ltd., and National Space Research and Development Agency, Nigeria. He is currently a Postdoctoral Fellow with Aalborg University, Denmark. His research interests include channel characterization, machine learning and AI for communications, intelligent spectrum access, and interference management.

TROELS PEDERSEN (Member, IEEE) received the M.Sc. degree in digital communications and the Ph.D. degree in wireless communications from Aalborg University, Aalborg, Denmark, in 2004 and 2009, respectively. In 2005, he was a Guest Researcher with FTW Telecommunications Research Center Vienna, Vienna, Austria. In 2009, he joined the Department of Electronic Systems, Aalborg University as an Assistant Professor, and became an Associate Professor in 2012. In 2012, he was a Visiting Professor with the Institut d'électronique et de Telecommunications de Rennes, University of Rennes 1, Rennes, France. His current research interests include statistical signal processing and communication theory, including sensor array signal processing, radio geolocation techniques, radio channel modeling, and radio channel sounding. He received the Teacher of the Year Award from the Study Board for Electronics and IT, Aalborg University, in 2011 and 2017.