



**Aalborg Universitet**

**AALBORG UNIVERSITY**  
DENMARK

## **Auto-Generated Summaries for Stochastic Radio Channel Models**

Bharti, Ayush; Adeogun, Ramoni Ojekunle; Pedersen, Troels

*Published in:*  
European Conference on Antennas and Propagation 2021

*Publication date:*  
2020

*Document Version*  
Early version, also known as pre-print

[Link to publication from Aalborg University](#)

*Citation for published version (APA):*  
Bharti, A., Adeogun, R. O., & Pedersen, T. (2020). Auto-Generated Summaries for Stochastic Radio Channel Models. Manuscript submitted for publication. In *European Conference on Antennas and Propagation 2021*

### **General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- ? Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- ? You may not further distribute the material or use it for any profit-making activity or commercial gain
- ? You may freely distribute the URL identifying the publication in the public portal ?

### **Take down policy**

If you believe that this document breaches copyright please contact us at [vbn@aub.aau.dk](mailto:vbn@aub.aau.dk) providing details, and we will remove access to the work immediately and investigate your claim.

# Auto-Generated Summaries for Stochastic Radio Channel Models

Ayush Bharti, Ramoni Adeogun, Troels Pedersen  
Department of Electronic Systems, Aalborg University, Denmark  
e-mail: {ayb, ra, troels}@es.aau.dk

**Abstract**—Recently, a calibration method has been proposed for estimating the parameters of stochastic radio channel models using summaries of channel impulse response measurements without multipath extraction. In this paper, we attempt to automatically generate summaries using an autoencoder for calibration of channel models. This approach avoids the need for explicitly designing informative summaries about the model parameters, which can be tedious. We test the method by calibrating the stochastic polarized propagation graph model on simulated as well as measured data. The autoencoder is found to generate summaries that give reasonably accurate results while calibrating the considered model.

**Index Terms**—radio channel modeling, autoencoder, approximate Bayesian computation, propagation graph, parameter estimation, machine learning.

## I. INTRODUCTION

Stochastic radio channel models are widely used for simulating the channel in order to design and test communication and localization systems. To ensure that such models yield accurate simulations, they must be calibrated. Most often, this is done by estimating the model parameters from measurement data. Unfortunately, the likelihood function for stochastic channel models are usually intractable. Thus, calibrating them from new measurements becomes challenging. Hence, it is standard practice to employ high-resolution path extraction algorithms to estimate the delays, gains, etc. of the multipath components. These estimates are further used to estimate the parameters of the channel model. This methodology has been followed to calibrate channel models from the early days of Turin [1] and Saleh-Valenzuela [2] to the more recent ones [3]–[5]. However, implementing such complicated algorithms is not trivial and requires a number of heuristic choices to be made. Such choices affect the accuracy of the results, and the overall performance of the estimator is difficult to assess.

Recently, calibration methods which circumvent the need for resolving the multipath components have been introduced [6]–[10]. In [6], [7], a method of moments approach is used for calibration. However, these methods rely on analytical expressions for the moments which may not be available for more complicated stochastic models. More general calibration methods based on summary statistics of the channel measurements have been proposed for the Saleh-Valenzuela model in [8], [9] and for the propagation graph model [11] in [10]. The methods proposed in [8], [10] are based on approximate Bayesian computation (ABC), which is a likelihood-free inference framework that relies on simulations from the model to

infer on the model parameters [12]. ABC involves comparing summary statistics of the simulated and the measured data in some distance metric. The parameter samples that yield simulated summaries “close” to the measured summaries are accepted. These accepted parameters form a sample from the approximate posterior distribution. Thus, such methods rely on handcrafted summaries of the measurement data which should be informative about the parameters of the stochastic channel models. However, designing such summaries is a time-consuming process which is not always straightforward for most models.

In this paper, we attempt to automatically generate the summaries by using an autoencoder [13]. An autoencoder is a combination of two neural networks; one encodes the input data-set into a low-dimensional set of features, and the other decodes those features in order to replicate the input data. We use the encoded feature vector as summary statistics to calibrate the polarized propagation graph model [11], [14] using the ABC algorithm [10]. This approach circumvents the need for manual design of summaries which can be a time-consuming process. We test the method by calibrating the stochastic polarized propagation graph model [11], [14] and comparing to previously obtained results for the algorithm [10] with handcrafted summaries. We find that the method requires much less effort than handcrafted summaries, while obtaining a comparable performance.

## II. ABC USING AUTOENCODER

We aim to fit a stochastic radio channel model,  $\mathcal{M}(\theta)$ , to a set of measurement data,  $\mathbf{x}$ . This amounts to estimating the  $p$ -dimensional parameter vector,  $\theta \in \mathbb{R}^p$  from the data. The ABC method in [10] allows us to do so provided simulations can be obtained from  $\mathcal{M}(\theta)$ . The ABC method relies on summarizing  $\mathbf{x}$  into a low-dimensional vector of summaries,  $\mathbf{s} = \mathcal{E}(\mathbf{x})$ , informative about  $\theta$ . In [8], [10], we relied on handcrafted summaries. Here, we circumvent designing the summarizing function, or encoder,  $\mathcal{E}(\cdot)$  ourselves by using an autoencoder to automatically learn this function.

### A. Generating Summaries using Autoencoder

As in Fig. 1, a typical autoencoder comprises of two neural networks — an encoder  $\mathcal{E}$  and a decoder  $\mathcal{D}$ , and an intermediate layer often referred to as the latent space. The encoder converts high-dimensional input variables into low-dimensional latent variables,  $\mathbf{s} = \mathcal{E}(\mathbf{x}) \in \mathbb{R}^q; q \ll n$ . The

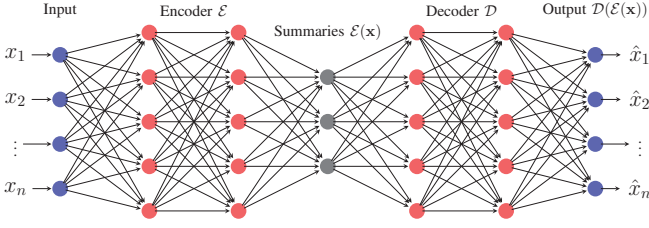


Fig. 1. Illustration of a deep autoencoder with 2 hidden layers in both the encoder and decoder.

decoder  $\mathcal{D}$  reconstructs the input from the latent variables. The goal is to obtain functions  $\mathcal{E}$  and  $\mathcal{D}$  such that the output,  $\hat{\mathbf{x}}$ , is close to the input in some metric, i.e.,

$$\hat{\mathbf{x}} = \mathcal{D}(\mathcal{E}(\mathbf{x})) \approx \mathbf{x}. \quad (1)$$

For the considered autoencoder architecture in Fig. 1, the encoding function reads

$$\mathbf{s} = \mathcal{E}(\mathbf{x}) = \eta_2(\mathbf{W}_2(\eta_1(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2)), \quad (2)$$

where  $\eta_i$ ,  $\mathbf{W}_i$  and  $\mathbf{b}_i$  denote the activation function, the weights, and the biases of the  $i^{\text{th}}$  hidden layer of the encoder, respectively. The decoding function,  $\mathcal{D}$ , is defined analogous to (2). The weights and biases are obtained by minimizing a reconstruction loss defined as the mean squared error between  $\mathbf{x}$  and  $\hat{\mathbf{x}}$ . Standard packages for performing such optimization exist in languages such as MATLAB, R and Python.

### B. Approximate Bayesian Computation method

The summarizing function  $\mathcal{E}$  obtained by the autoencoder is now used in the ABC algorithm proposed in [10] to approximate the posterior distribution,  $p(\boldsymbol{\theta}|\mathbf{s}_{\text{obs}})$ , where  $\mathbf{s}_{\text{obs}}$  is the summary vector of the measurements. The algorithm is stated in Alg. 2 and illustrated in Fig. 2. Here, we give an overview of the ABC algorithm; see [10] for further details.

The ABC method proceeds by sampling  $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_M$  independently from the prior distribution  $p(\boldsymbol{\theta})$ , and simulating the corresponding summaries  $\mathbf{s}_1, \dots, \mathbf{s}_M$  using the model and the summarizing function. The Euclidean distance between the simulated and the observed statistics,  $\|\mathbf{s}_i - \mathbf{s}_{\text{obs}}\|$ , are then computed. Note that the summaries are normalized using their mean absolute deviation before computing the distance. The first  $M_\epsilon$  parameter samples that correspond to the smallest distance are accepted, along with their corresponding summary vectors. This results in an acceptance ratio of  $\epsilon = M_\epsilon/M$ . The accepted samples are then adjusted using local-linear regression [15] to improve the posterior approximation. Given the accepted set  $\{(\mathbf{s}_i, \boldsymbol{\theta}_i)\}_{i=1}^{M_\epsilon}$ , the  $i^{\text{th}}$  accepted parameter sample is adjusted as

$$\tilde{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_i - (\mathbf{s}_i - \mathbf{s}_{\text{obs}})^T \hat{\boldsymbol{\beta}}, \quad i = 1, \dots, M_\epsilon, \quad (3)$$

where  $\hat{\boldsymbol{\beta}}$  is solution to the optimization problem

$$\arg \min_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \sum_{i=1}^{M_\epsilon} \left[ \boldsymbol{\theta}_i - \boldsymbol{\alpha} - (\mathbf{s}_i - \mathbf{s}_{\text{obs}})^T \boldsymbol{\beta} \right]^2 K_\epsilon(\|\mathbf{s}_i - \mathbf{s}_{\text{obs}}\|). \quad (4)$$

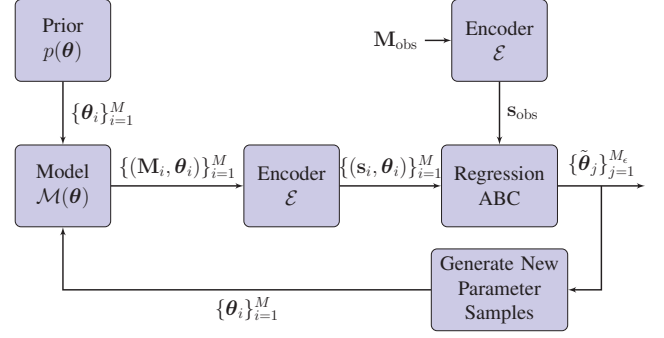


Fig. 2. Block diagram of the data flow in the proposed PMC-ABC algorithm with regression adjustment.

### Algorithm 1 Regression adjustment

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

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

Solve optimisation problem (4) with  $\{\boldsymbol{\theta}_j^*\}_{j=1}^{M_\epsilon}$  and corresponding  $\{\mathbf{s}_j^*\}_{j=1}^{M_\epsilon}$  to get  $\hat{\boldsymbol{\beta}}$

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

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

Here,  $K_\epsilon(\cdot)$  is the Epanechnikov kernel. The ABC method of [10] then draws a new set of  $M$  parameter samples from  $(\tilde{\boldsymbol{\theta}}_1, \dots, \tilde{\boldsymbol{\theta}}_{M_\epsilon})$  in a sequential Monte Carlo fashion [16]. These new samples form the prior distribution for the next iteration of the algorithm, where they are used to generate simulated data from the model again and perform regression adjustment. The idea is to iteratively converge towards the posterior distribution by sampling the parameter space efficiently. The parameter samples for iteration  $t$  are drawn from the density kernel

$$\varphi_t(\boldsymbol{\theta}) = \sum_{j=1}^{M_\epsilon} \mathbf{w}_j^{(t-1)} \mathcal{K} \left( \boldsymbol{\theta}^{(t)} | \tilde{\boldsymbol{\theta}}_j^{(t-1)}; \boldsymbol{\sigma}_{(t-1)}^2 \right), \quad (5)$$

where  $\mathbf{w}_j^{(t-1)}$  is the importance sampling weight associated with  $\tilde{\boldsymbol{\theta}}_j^{(t-1)}$ , and  $\boldsymbol{\sigma}_{(t-1)}^2$  is the variance vector with each entry associated with the Gaussian kernel,  $\mathcal{K}$ , truncated to be in the prior range. The adjusted parameter samples after  $N_{\text{iter}}$  iterations are then taken as samples from the approximate posterior distribution.

## III. CALIBRATION OF STOCHASTIC CHANNEL MODELS

### A. Calibration Problem

We apply the calibration method to calibrate the stochastic polarized propagation graph model (SPPGM) [11] in which the channel is represented as a propagation graph [17] with the transmitters, receivers and scatterers as vertices. Edges in the graph are defined randomly depending on the probability of visibility,  $P_{\text{vis}}$ . An edge transfer function accounting for depolarization effects, attenuation, delay and phase shifts is defined for each edge. To calculate the edge transfer functions,

---

**Algorithm 2** ABC method [10]**Input:** Prior  $p(\theta)$ , model  $\mathcal{M}(\theta)$ , observed summaries  $\mathbf{s}_{\text{obs}}$ ,  $M_\epsilon$ ,  $M$ , $N_{\text{iter}}$ Initialization:  $t = 1$ ,**for**  $i = 1$  to  $M$  **do**Sample  $\theta_i^{(1)} \sim p(\theta)$ Simulate  $\mathbf{M}_i^{(1)} \sim \mathcal{M}(\theta_i^{(1)})$  and compute  $\mathbf{s}_i^{(1)} = \mathcal{E}(\mathbf{x}_i^{(1)})$ **end for**Perform regression adjustment by applying **Algorithm 1** on $\left\{ \left( \mathbf{s}_i^{(1)}, \theta_i^{(1)} \right) \right\}_{i=1}^M$  to obtain  $\left\{ \tilde{\theta}_j^{(1)} \right\}_{j=1}^{M_\epsilon}$ 

Set weights

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

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

**for**  $t = 2$  to  $N_{\text{iter}}$  **do****for**  $i = 1, \dots, M$  **do**Sample  $\theta_i^* \sim \left\{ \tilde{\theta}_j^{(t-1)} \right\}_{j=1}^{M_\epsilon}$  with probabilities  $\mathbf{w}_j^{(t-1)}$ Generate  $\theta_i^{(t)} \sim \mathcal{K}_t(\theta | \theta_i^*; \sigma_{(t-1)}^2)$ Simulate  $\mathbf{M}_i^{(t)} \sim \mathcal{M}(\theta_i^{(t)})$  and compute  $\mathbf{s}_i^{(t)} = \mathcal{E}(\mathbf{x}_i^{(t)})$ **end for**Perform regression adjustment by applying **Algorithm 1** on $\left\{ \left( \mathbf{s}_i^{(t)}, \theta_i^{(t)} \right) \right\}_{i=1}^M$  to obtain  $\left\{ \tilde{\theta}_j^{(t)} \right\}_{j=1}^{M_\epsilon}$ 

Set weights

$$\mathbf{w}_j^{(t)} \propto \frac{p(\theta_j^{(t)})}{\sum_{j=1}^{M_\epsilon} \mathbf{w}_j^{(t-1)} \mathcal{K}_t(\theta_j^{(t)} | \tilde{\theta}_j^{(t-1)}; \sigma_{(t-1)}^2)}, \quad 1 \leq j \leq M_\epsilon$$

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

**end for****Output:** Samples  $(\tilde{\theta}_1^{(T)}, \dots, \tilde{\theta}_{M_\epsilon}^{(T)})$  from the approximate posterior

the SPPGM only requires three parameters viz: reflection coefficient  $g$ , number of scatterers  $N_s$  and the polarization ratio  $\gamma$ . The edge transfer functions are then used in a simple expression to compute the channel transfer function,  $H_k$ . Detailed description of the model and channel generation procedure can be found in [11].

We consider data from a linear, time-invariant radio channel, measured using a vector network analyzer (VNA) in the bandwidth  $B$ . The transfer function  $H_k$  is measured at  $K$  equidistant frequency points resulting in a frequency separation of  $\Delta f = B/(K-1)$ . The measured signal at each frequency point,  $Y_k$ , can be modeled as

$$Y_k = H_k + N_k, \quad k = 0, 1, \dots, K-1, \quad (6)$$

where  $N_k$  denotes the measurement noise. The noise samples at each  $k$  are assumed independent and identically distributed (iid) as a circular complex Gaussian with variance  $\sigma_N^2$ . Discrete-frequency, continuous-time inverse Fourier trans-

forming  $\{Y_k\}_{k=0}^{K-1}$  gives the measured signal in time-domain

$$y(t) = \frac{1}{K} \sum_{k=0}^{K-1} Y_k \exp(j2\pi k \Delta f t), \quad (7)$$

which is periodic with period  $t_{\text{max}} = 1/\Delta f$ . Typically  $K$  is in order of hundreds or even thousands, and so we intend to summarize the high-dimensional measured signal into its first  $J$  temporal moments, defined as

$$m_j = \int_0^{t_{\text{max}}} t^j |y(t)|^2 dt, \quad j = 0, 1, 2, \dots, (J-1). \quad (8)$$

The temporal moments are computed instantaneously per realization of  $y(t)$ . For the dual polarized channel from the SPPGM, this computation is performed for each of the four polarizations. Thus the  $i^{\text{th}}$  realization yields a  $4J$  dimensional vector  $\mathbf{m}^{(i)}$ . Consequently, a measurement with  $L$  independent polarimetric measurements yields an  $L \times 4J$  matrix of temporal moments  $\mathbf{M}_{\text{obs}} = [\mathbf{m}^{(1)}, \dots, \mathbf{m}^{(L)}]^T$ . Including the noise variance as a parameter, calibration of the SPPGM therefore requires estimating the parameter vector  $\theta = [g, N_s, P_{\text{vis}}, \gamma, \sigma_N^2]^T$ .

**B. Autoencoder Implementation and Training**

We consider the first three temporal moments,  $J = 3$ , and  $L = 625$  realizations. The training data is obtained from the SPPGM with 4000 parameter vectors generated uniformly over the prior ranges in Tab. I. For each parameter vector,  $\theta_i$ , we compute the temporal moments using (8) and convert the  $625 \times 12$  matrix  $\mathbf{M}_i$  into an input vector,  $\mathbf{x}_i \in \mathbb{R}^{7500}$ . To be consistent with [10], we set the number of summaries  $q = 10$ .

We adopt Python's popular machine learning libraries, *Keras* and *Tensorflow*, to design and implement the autoencoder. We use the Rectified Linear Unit activation function at the hidden layers of both the encoder and decoder due to its non-vanishing gradient property. A linear activation function is used at the output layer of the decoder. Based on our initial experiments, we observe that a network with two hidden layers in the encoder and decoder yields reasonably informative summaries. We adopt a mini-batch gradient descent procedure in which the training data is partitioned into equal batches of size 32. The training is performed using the Root Mean Square Propagation (RMSProp) algorithm with learning rate = 0.1 as optimizer.

**C. Evaluation of summaries**

We now use the trained encoder to test whether the auto-generated summary vector  $\mathbf{s}$  is informative about the parameters of the SPPGM. We do that by varying one parameter at a time while setting others to a fixed value, and computing  $\mathbf{s}$ . If the summaries have a functional relationship with the parameters, then they are deemed informative. Each parameter is varied in its prior range, and the other parameters are fixed to true values given in Tab. I. The resulting plot for  $q = 10$  summaries is shown in Fig. 3. We see that the summaries are responsive to changes in each of the parameters, albeit more for some than others. The generated summaries

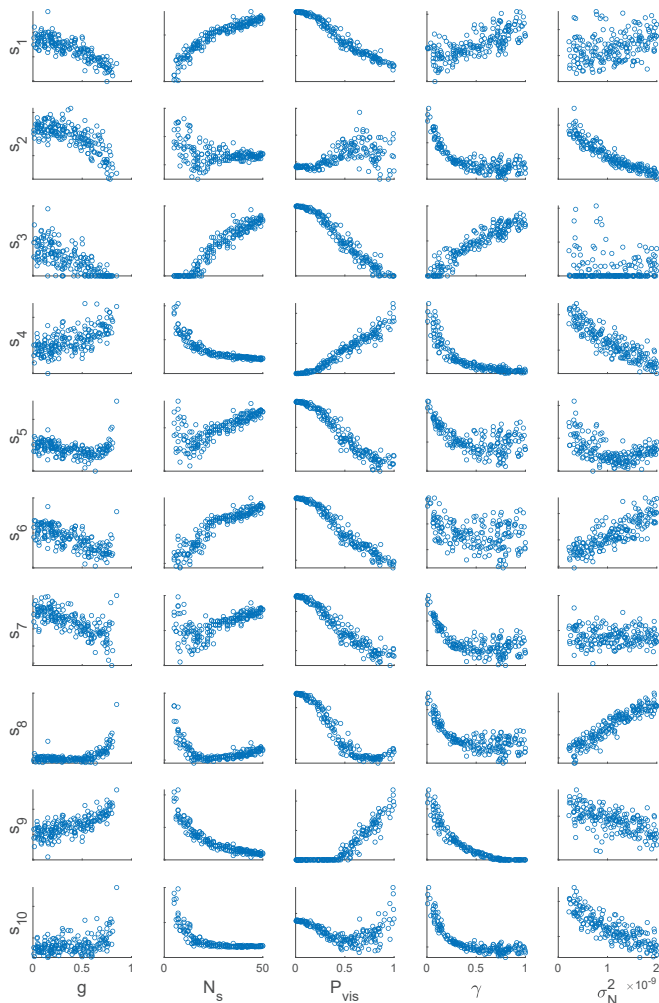


Fig. 3. Auto-generated statistics versus the parameters of the SPPGM. Each plot is generated by varying one parameter while others are kept fixed to the true values in Tab. I.

are informative enough to be able to calibrate the SPPGM. However, this visual test gives only a picture of how well the summaries work separately. To evaluate how informative the joint summaries are about the parameters, we apply these in the ABC algorithm.

#### D. Application to simulated data

We apply Alg. 2 to calibrate the SPPGM from simulated data. We set the parameter vector to a true value and generate polarized channel measurements from the SPPGM which we then summarize using the encoder to get the summaries. We take an average over 200 realizations of such summaries to get  $s_{\text{obs}}$  in order to remove any bias in the estimate arising due to Monte Carlo approximation. The prior distributions were kept same as in [10] and are given in Tab. I, along with the approximate minimum mean squared error (MMSE) estimates. The approximate posterior distributions are shown in Fig. 4. We observe that the marginal posteriors are concentrated around the true values, and that the algorithm seems to work. The width of the posteriors indicate how informative the

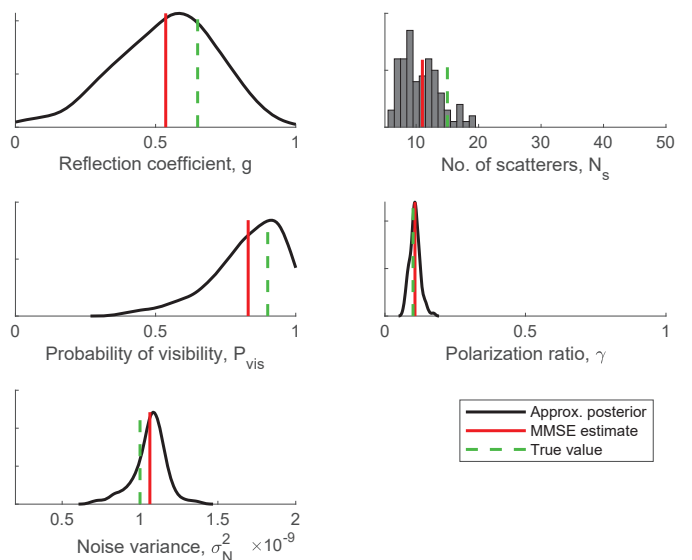


Fig. 4. Kernel density estimates of the approximate marginal posteriors for simulated data after  $N_{\text{iter}} = 10$  iteration. Settings:  $L = 625$ ,  $J = 3$ ,  $M = 2000$ ,  $M_{\epsilon} = 100$ ,  $B = 4$  GHz,  $K = 801$ ,  $\Delta f = 5$  MHz,  $t_{\text{max}} = 200$  ns.

TABLE I  
PARAMETER ESTIMATES FOR SIMULATED AND MEASURED DATA. THE SAMPLE STANDARD DEVIATION OF THE APPROXIMATE POSTERIOR IS REPORTED IN THE PARENTHESIS.

Parameter $\theta$	Prior range $p(\theta)$	True value / Estimate (standard deviation)	
		Simulated data	Measured data
Refl. coeff. $g$	[0,1]	0.65 / 0.54 (0.18)	- / 0.71 (0.15)
No. of scat. $N_s$	[5,50]	15 / 11 (3.08)	- / 16 (5.48)
Prob. of vis. $P_{\text{vis}}$	[0,1]	0.90 / 0.83 (0.13)	- / 0.70 (0.14)
Pol. ratio $\gamma$	[0,1]	0.10 / 0.11 (0.02)	- / 0.14 (0.02)
Noise variance $\sigma_N^2 \times 10^{-10}$	[2,20]	10.0 / 10.6 (1.07)	- / 3.75 (0.51)

summarise are about each parameter. For example, the fact that the posterior for  $g$  is the widest is due to its lack of any distinct relationship with statistics in Fig. 3. Overall the method seems to work reasonably well, considering that no manual effort went into designing the specific summaries.

#### IV. APPLICATION TO MEASURED DATA

We now apply the method to calibrate the SPPGM using millimetre-wave polarized channel measurements from [18]. The measurements were conducted using dual-polarized antennas in a small conference room of dimensions  $3 \times 4 \times 3$  m<sup>3</sup> in the frequency range 58 GHz to 62 GHz. A  $5 \times 5$  virtual planar array was used at both the transmitter and receiver with 5 mm inter-element spacing, resulting in  $L = 625$  realizations for each polarization.

The approximate marginal posteriors are given in Fig. 5 while the approximate MMSE estimates, obtained by averaging the accepted samples after the last iteration, are reported in Tab. I. The width of the posterior estimates are similar to what we observed in simulation. The summaries therefore seem to be useful for calibration even in case of measurements. The averaged power delay profile (APDP) of the co- and cross-polarized channel from the measurements is compared to that

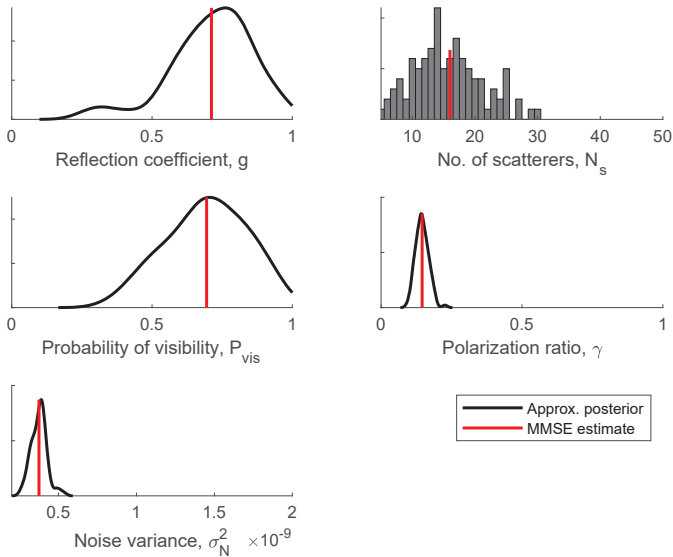


Fig. 5. Kernel density estimates of the approximate marginal posteriors for measured data after 10th iteration. The algorithm settings are same as given in Fig. 4.

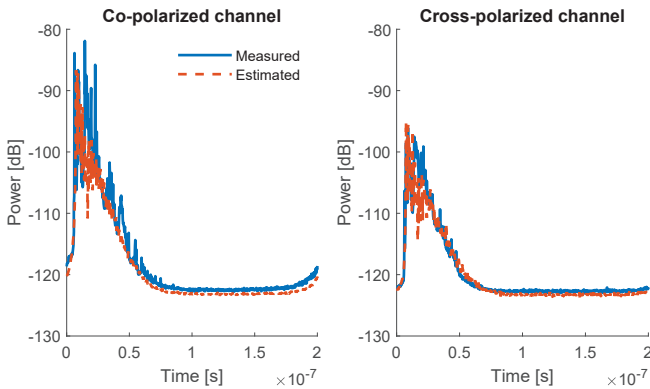


Fig. 6. Averaged power delay profile from the measurements versus that obtained from the SPPGM after calibration. The parameter estimates are reported in Tab. I.

from the SPPGM in Fig. 6. The estimated APDP seems to fit the measurements well, thus validating the method.

## V. CONCLUSIONS

The autoencoder is able to generate summaries that are informative enough to calibrate the parameters of the polarized propagation graph model. In this case, handcrafting of summaries is not necessary for implementing the calibration method of [10]. Avoiding the need for handcrafted summaries enables even non-propagation experts to easily apply the method. Even though the performance of the calibration method is better when using the handcrafted summaries in [10], we do get reasonably accurate results with fairly limited effort of training the autoencoder. However, since the training is done in an unsupervised manner, there is no guarantee that the generated summaries will be informative enough about the

parameters to be able to estimate them. With auto-generated summaries, we are one step closer to fully automated model calibration.

## ACKNOWLEDGMENT

The authors would like to thank Dr. Carl Gustafson and Prof. Fredrik Tufvesson (Lund University) for providing the measurement data. This work is supported by the Danish Council for Independent Research, grant no. DFF 7017-00265. Ramoni Adeogun is supported by grant no. DFF 9041-00146B.

## REFERENCES

- [1] 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, pp. 1–9, Feb 1972.
- [2] A. A. M. Saleh and R. Valenzuela, "A statistical model for indoor multipath propagation," *IEEE J. Sel. Areas Commun.*, vol. 5, pp. 128–137, February 1987.
- [3] P. Kyösti, *WINNER II channel models, deliverables D1.1.2 V1.2, part I: Channel models*.
- [4] 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, 2014.
- [5] J. Li, B. Ai, R. He, M. Yang, Z. Zhong, and Y. Hao, "A cluster-based channel model for massive mimo communications in indoor hotspot scenarios," *IEEE Trans. on Wireless Commun.*, vol. 18, no. 8, pp. 3856–3870, 2019.
- [6] W.-D. Wu, C.-H. Wang, C.-C. Chao, and K. Witrisal, "On parameter estimation for ultra-wideband channels with clustering phenomenon," in *2008 IEEE 68th Veh. Technol. Conf.*, IEEE, Sep 2008.
- [7] A. Bharti, R. Adeogun, and T. Pedersen, "Estimator for Stochastic Channel Model without Multipath Extraction using Temporal Moments," in *20th IEEE Int. Workshop on Signal Process. Advances in Wireless Commun. (SPAWC)*, 2019.
- [8] A. Bharti and T. Pedersen, "Calibration of stochastic channel models using approximate Bayesian computation," in *Proc. IEEE Global Commun. Conf. Workshops*, 2019.
- [9] R. Adeogun, "Calibration of stochastic radio propagation models using machine learning," *IEEE Antennas and Wireless Propag. Letters*, vol. 18, pp. 2538–2542, Dec 2019.
- [10] A. Bharti, R. Adeogun, and T. Pedersen, "Learning parameters of stochastic radio channel models from summaries," *IEEE Open J. of Antennas and Propag.*, pp. 1–1, 2020.
- [11] R. Adeogun, T. Pedersen, C. Gustafson, and F. Tufvesson, "Polarimetric Wireless Indoor Channel Modelling Based on Propagation Graph," *IEEE Trans. on Antennas and Propag.*, vol. 67, no. 10, 2019.
- [12] J. Lintusaari, M. U. Gutmann, R. Dutta, S. Kaski, and J. Corander, "Fundamentals and recent developments in approximate bayesian computation," *Systematic Biology*, p. syw077, Oct 2016.
- [13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [14] R. Adeogun and T. Pedersen, "Propagation graph based model for multipolarized wireless channels," in *IEEE WCNC*, April 2018.
- [15] M. A. Beaumont, W. Zhang, and D. J. Balding, "Approximate bayesian computation in population genetics," *Genetics*, vol. 162, no. 4, pp. 2025–2035, 2002.
- [16] M. A. Beaumont, J.-M. Cornuet, J.-M. Marin, and C. P. Robert, "Adaptive approximate bayesian computation," *Biometrika*, vol. 96, pp. 983–990, oct 2009.
- [17] T. Pedersen, G. Steinböck, and B. H. Fleury, "Modeling of reverberant radio channels using propagation graphs," vol. 60, pp. 5978–5988, Dec 2012.
- [18] C. Gustafson, D. Bolin, and F. Tufvesson, "Modeling the polarimetric mm-wave propagation channel using censored measurements," in *2016 Global Commun. Conf.*, IEEE, Dec 2016.