NUCLEAR SCIENCE AND ENGINEERING · VOLUME 197 · 1928–1951 · AUGUST 2023

DOI: https://doi.org/10.1080/00295639.2022.2143705

Multi-Output Gaussian Processes for Inverse Uncertainty Quantification in Neutron Noise Analysis

Paul Lartaud,[a]* Philippe Humbert,[a] and and Josselin Garnier[b]

[a]*Commissariat à l'Energie Atomique (CEA), Arpajon, France*
[b]*Centre de Mathématiques Appliquées (CMAP), Ecole polytechnique, Institute Polytechnique de Paris, Paris, France*

Received August 8, 2022
Accepted for Publication November 1, 2022

Abstract — *In a fissile material, the inherent multiplicity of neutrons born through induced fissions leads to correlations in their detection statistics. The correlations between neutrons can be used to trace back some characteristics of the fissile material. This technique, known as neutron noise analysis, has applications in nuclear safeguards or waste identification. It provides a nondestructive examination method for an unknown fissile material. This is an example of an inverse problem where the cause is inferred from observations of the consequences.*

However, neutron correlation measurements are often noisy because of the stochastic nature of the underlying processes. This makes the resolution of the inverse problem more complex since the measurements are strongly dependent on the material characteristics. A minor change in the material properties can lead to very different outputs. Such an inverse problem is said to be ill posed. For an ill-posed inverse problem, the inverse uncertainty quantification is crucial. Indeed, seemingly low noise in the data can lead to strong uncertainties in the estimation of the material properties. Moreover, the analytical framework commonly used to describe neutron correlations relies on strong physical assumptions, and is thus inherently biased.

This paper addresses dual goals. First, surrogate models are used to improve neutron correlation predictions and quantify the errors on those predictions. Then the inverse uncertainty quantification is performed to include the impact of measurement error alongside the residual model bias.

Keywords — *Neutron noise analysis, uncertainty quantification, nuclear safeguards, supervised learning, Gaussian processes.*

Note — *Some figures may be in color only in the electronic version.*

I. INTRODUCTION

Neutron noise analysis describes a set of methods that aims to identify a fissile material based on observations of neutron correlations in the multiplying medium.[1] These methods can have applications in nuclear safeguards, criticality accident detections, or waste identification.[2] Neutron correlations can be described analytically within a simplified framework known as the point model approximation.[3] This model can be used to solve an inverse problem and evaluate characteristics of the medium, such as its prompt multiplication. However, the point model relies on strong physical assumptions, which leads to a systematic error in the predictions. The bias introduced by the point model is often disregarded, and the uncertainty quantification of neutron noise techniques is not often considered.

The objective of this paper is to provide a robust inverse uncertainty quantification method that accounts for the bias introduced by the point model. The inverse problem is solved with a Bayesian approach as is done in several works in stochastic neutronics.[4,5] Surrogate models are then used to replace and generalize the point model. The surrogate models are based on Gaussian

*E-mail: paul.lartaud@polytechnique.edu

processes,[6] which are a flexible nonparametric regression tool. The surrogate models provide better predictions of neutron correlations and are computationally cheap to call. On top of this, they also yield covariances of the predictions. The predicted covariances can be introduced into the Bayesian resolution of the inverse problem to include the model uncertainties into the estimation of the posterior distribution of the input parameters. From the results obtained, the method presented in this paper is able to significantly improve neutron noise techniques while simultaneously providing a robust inverse uncertainty quantification.

This paper provides a brief description of the point model and presents a summary of Gaussian process regression (GPR) and its extension to multi-output problems. Then the general methodology is presented and tested on two different examples extracted from neutron multiplicity experiments.

II. POINT MODEL FRAMEWORK

In this section, a brief introduction to the point model approximation in neutron noise theory is presented.

II.A. Neutron Correlations

In a fissile material, one neutron can induce a fission, leading to the birth of more neutrons and so on. This process creates correlations between the neutrons. The successive fissions can be described as branching processes, where each fission event is a node in a fission chain leading to the creation of more branches.

Let us consider a setup with a fissile material and a neutron detector. At some point, two neutron detections can be recorded simultaneously in the detector, or more specifically, within the same time gate of temporal size T. This simultaneous double detection can either be accidental if the neutrons are independent, or correlated if they are not. Two neutrons are correlated if and only if they belong to the same fission chain. With this criterion, it is possible to make a distinction between true correlated double detections and accidental double detections. Neutron noise techniques study the occurrence of true correlated double and triple detections in the detectors.

II.B. Point Model Assumptions

The following assumptions are made in the point model framework:

1. The medium is infinite, homogeneous, isotropic, and subcritical.

2. Only fission and capture reactions occur in the material. They are governed, respectively, by the macroscopic cross sections Σ_f and Σ_c.

3. Neutrons are monoenergetic.

4. The source is either a spontaneous fission source, an (α, n) source, or a mix of the two.

5. Neutrons are detected by a neutron capture such that the total macroscopic capture cross section can be written $\Sigma_c = \Sigma_d + \Sigma_p$ where Σ_d is the detection cross section and Σ_p is the parasitic capture cross section.

These assumptions largely simplify the description of neutron correlations. An analytical description of neutron correlations is possible in this framework. The derivation of the point model equations is not provided here. Depending on the inputs and outputs considered, the point model equations can be found under different forms. Most of the time, Böhnel equations are used, but in this paper the Feynman/Furuhashi framework is studied instead.

II.C. Böhnel Equations

Böhnel equations establish a link between the material characteristics and neutron correlations. The outputs investigated are the single count rate C_1, the correlated double-detection count rate C_2, and the correlated triple-detection count rate C_3.

The point model equations depend on the distribution of neutron multiplicity in induced and spontaneous fissions. The average number of neutrons produced per induced fissions is denoted $\overline{\nu}$. The second- and third-order factorial moments of the multiplicity distribution $\overline{\nu}_2 = \overline{\nu(\nu-1)}$ and $\overline{\nu}_3 = \overline{\nu(\nu-1)(\nu-2)}$ are introduced. The average number of neutrons for spontaneous fissions $\overline{\nu}_s$ and the factorial moments for spontaneous fissions $\overline{\nu}_{2s}$ and $\overline{\nu}_{3s}$ are defined similarly.

The input parameters in the Böhnel framework are the leakage multiplication M_L, the leakage efficiency ε_L defined as the number of counts per nonfissioning neutrons, and the fission source intensity Q_F, which is the number of spontaneous fission source events per second. Finally, the alpha ratio $\alpha_R = \frac{\overline{\nu}_s Q_F}{Q_\alpha}$ is the ratio of source neutrons produced by spontaneous fissions over the number of neutrons produced by (α, n).

Based on the assumptions described in the previous sections, the point model equations can be derived as

$$C_1 = \varepsilon_L M_L \overline{\nu}_s Q_F (1 + \alpha_R) \ , \tag{1}$$

$$C_2 = \frac{\varepsilon_L^2 M_L^2 Q_F}{2} \left(\overline{\nu}_{2s} + \overline{\nu}_2 \overline{\nu}_s (1 + \alpha_R) \left(\frac{M_L - 1}{\overline{\nu} - 1} \right) \right) , \tag{2}$$

and

$$C_3 = \frac{\varepsilon_L^3 M_L^3 Q_F}{6} \overline{\nu}_{3s} + 3(1 + \alpha_R) \overline{\nu}_s \overline{\nu}_2^2 \left(\frac{M_L - 1}{\overline{\nu} - 1} \right)^2 + \left((1 + \alpha_R) \overline{\nu}_3 \overline{\nu}_s + 3 \overline{\nu}_{2s} \overline{\nu}_2 \right) \left(\frac{M_L - 1}{\overline{\nu} - 1} \right) . \tag{3}$$

A detailed derivation of the equations can be found in Ref. 7.

II.D. Feynman Equations

In the Feynman/Furuhashi framework,[8] the inputs considered are the prompt multiplication factor k_p, the detector efficiency (or Feynman efficiency) ε_F defined as the number of counts per induced fissions, the source intensity S defined as the number of source events (spontaneous fission or (α, n) reaction) per second, and $x_s = (1 + \alpha_R)^{-1}$, which is the ratio of source neutrons produced per spontaneous fissions over the total number of source neutrons.

The Diven factors of second and third order D_2 and D_3 for induced fissions are also introduced. They are defined as the second- and third-order reduced factorial moments of the multiplicity distribution,

$$D_2 = \frac{\overline{\nu(\nu - 1)}}{\overline{\nu}^2} = \frac{\overline{\nu}_2}{\overline{\nu}^2}$$

and

$$D_3 = \frac{\overline{\nu(\nu - 1)(\nu - 2)}}{\overline{\nu}^3} = \frac{\overline{\nu}_3}{\overline{\nu}^3} \ . \tag{4}$$

The Diven factors for the spontaneous fission multiplicity D_{2s} and D_{3s} are defined similarly.

The outputs of the Feynman model are the average count rate R and the second and third asymptotic Feynman moment Y_∞ and X_∞.

The second- and third-order Feynman moments $Y(T)$ and $X(T)$ are defined for a given size T of a detection window as

$$Y(T) = \frac{2 N_{2C}(T)}{N_1(T)}$$

and

$$X(T) = \frac{6 N_{3C}(T)}{N_1(T)} \ , \tag{5}$$

where $N_{2C}(T)$ and $N_{3C}(T)$ are, respectively, the average number of double and triple correlated detections in a time window of size T, and $N_1(T)$ is the average number of counts per time window. In this formalism, the average count rate is the average number of counts divided by the time width. It does not depend on T:

$$R = \frac{N_1(T)}{T} \ . \tag{6}$$

The Feynman moments depend on the size of the time window considered. However, for T large, $Y(T)$ [respectively, $X(T)$] converges toward an asymptotic value Y_∞ [respectively, X_∞]. Indeed, in a subcritical medium, the fission chains have a limited lifetime. For T much larger than the average fission chain lifetime, all the true correlations are encapsulated in the detection window. Thus, the Feynman moments reach an asymptotic value.

Based on the Böhnel equations, the equations for the Feynman/Furuhashi framework can be derived. The prompt reactivity $\rho = \frac{k_p - 1}{k_p} < 0$ is introduced as

$$R = \frac{\varepsilon_F S \, \overline{\nu}_s}{-\rho \overline{\nu} (\overline{\nu}_s + x_s - \overline{\nu}_s x_s)} \ , \tag{7}$$

$$Y_\infty = \frac{\varepsilon_F D_2}{\rho^2} \left(1 - x_s \rho \frac{\overline{\nu}_s D_{2s}}{\overline{\nu} D_2} \right) , \tag{8}$$

and

$$X_\infty = 3 \left(\frac{\varepsilon_F D_2}{\rho^2} \right)^2 \left(1 - x_s \rho \frac{\overline{\nu}_s D_{2s}}{\overline{\nu} D_2} \right) - \frac{\varepsilon_F^2 D_3}{\rho^3}^2 \left(1 - x_s \rho \frac{\overline{\nu}_s^2 D_{3s}}{\overline{\nu}^2 D_3} \right) . \tag{9}$$

II.E. Limitations

The point model framework is based on strong physical assumptions. It is expected to yield biased

predictions, especially in cases where strong heterogeneities are present in the medium.

A simple geometry is created to highlight this bias. Let us consider a metallic Pu spherical shell containing 94.0 at. % of ^{239}Pu and 6.0 at. % of ^{240}Pu, with at. % referring to the atomic percentage of the isotope in the material. The shell has a density of 19.6040 g·cm^{-3} with an internal and external diameter, respectively, of 7.014 and 12.208 cm. The internal region is void. The fissile region is surrounded by a borated polyethylene layer with a density of 1.00 g·cm^{-3} and 5.8 at. % of boron carbide B_4C, and a thickness of 3.024 cm.

The average count rate and the Feynman moments are evaluated based on the method described later in Sec. IV.B. Similarly, the input parameters $\theta = (k_p, \varepsilon_F, S, x_s)$ are recorded. Then, the point model predictions for the inputs θ are compared to the measured values of (R, Y_∞, X_∞). The results are summarized in Table I.

The point model does provide significant bias for all three outputs. The most prominent error is the error on the count rate R because this is the least noisy output. Consequently, in the inverse problem resolution, a bias on R is more penalizing than a bias on X_∞, which is significantly noisier and thus has a larger associated variance.

The large biases observed in the point model impact the inference of the input parameters in the inverse problem. For such ill-posed problems, this bias can lead to significant errors in the estimation of θ. For that reason, one of the objectives of this paper is to build a surrogate model able to improve predictions of (R, Y_∞, X_∞). The surrogate model should also include an estimation of the uncertainty of the predictions in order to include the residual model bias in the resolution of the inverse problem.

III. BUILDING A SURROGATE MODEL

Our objective is to replace the point model with a surrogate model. The surrogate model should be able to provide better predictions than the point model. For this purpose, more input parameters were considered when building the surrogate model in order to account for the neutron spectrum and parasitic absorption or leakage, which are not considered in the point model. A metric of the quality of the predictions is also expected from the surrogate model. Namely, the goal is to be able to provide a mean prediction for the output vector (R, Y_∞, X_∞) with its associated covariance.

The covariance predictions are meant to be used directly in the Monte Carlo Markov chain (MCMC) methods to sample the posterior distribution of the input vector θ [that may be $\theta = (k_p, \varepsilon_F, S, x_s)$ or that may include more input parameters]. The aim is to include the model uncertainty in the model parameters. In this work, Gaussian processes are used as surrogate models.

TABLE I

Point Model Bias for a Test Case with a Strong Heterogeneous Configuration

	R	Y_∞	X_∞
MCNP6 data	15 120	1.39	5.52
Point model	16 260	1.29	4.88
Relative error (%)	7.5	7.2	11.6

III.A. Scalar Gaussian Processes

III.A.1. Introductory Concepts

Gaussian process regression is a nonparametric Bayesian regression method. It is a flexible tool for regression that is able to quantify uncertainties in the predictions. In this section, a brief description of GPR is given. This is not meant to be a thorough guideline but rather an introduction for beginners.

Definition III.1

A Gaussian process is a collection of random variables, such that any finite subset follows a multivariate normal distribution. The distribution of a Gaussian process is completely defined by its mean function $m(\cdot)$ and covariance function $k(\cdot, \cdot)$. The Gaussian process formalizes the concept of distributions over functions.

Let us first consider scalar Gaussian processes with one real output and multidimensional inputs. The input dimension is I. If f is a Gaussian process with mean m and covariance function k, it is denoted as

$$f \sim \mathcal{GP}(m(x), k(x, x')) . \tag{10}$$

Assuming a Gaussian process model $f \sim \mathcal{GP}(m(x), k(x, x'))$ with scalar outputs, one can draw samples of the distribution of functions evaluated over N input points represented by the matrix $\mathbf{X} = (X_i)_{i \leq N} \in \mathbb{R}^{N \times I}$. This sample vector $\mathbf{f} \in \mathbb{R}^N$ follows a multivariate normal distribution,

$$\mathbf{f} \sim \mathcal{N}(m(\mathbf{X}), K(\mathbf{X},\mathbf{X})) \ , \tag{11}$$

where $m(\mathbf{X}) \in \mathbb{R}^N$ is the mean vector and $K(\mathbf{X},\mathbf{X}) \in \mathbb{R}^{N\times N}$ is the covariance matrix defined by $K(\mathbf{X},\mathbf{X})_{i,j} = k(X_i, X_j)$ for $i,j \leq N$.

III.A.2. Predictions with Gaussian Processes

Let us consider a Gaussian process $f \sim \mathcal{GP}(m(x), k(x,x'))$. Let $\mathbf{X} \in \mathbb{R}^{N\times I}$ and $\mathbf{y} \in \mathbb{R}^N$ be, respectively, N training inputs and outputs. Similarly, let $\mathbf{X}' \in \mathbb{R}^{N'\times I}$ and $\mathbf{y}' \in \mathbb{R}^{N'}$ be, respectively, N' test inputs and outputs.

Given the properties of Gaussian processes, the joint distribution of training and test outputs is

$$\begin{pmatrix} \mathbf{f} \\ \mathbf{f}' \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} m(\mathbf{X}) \\ m(\mathbf{X}') \end{pmatrix}, \begin{pmatrix} K(\mathbf{X},\mathbf{X}) & K(\mathbf{X},\mathbf{X}') \\ K(\mathbf{X}',\mathbf{X}) & K(\mathbf{X}',\mathbf{X}') \end{pmatrix} \right) . \tag{12}$$

The conditional distribution of the test outputs $\mathbf{f}'$ given $\mathbf{X}$, $\mathbf{X}'$, and $\mathbf{y}$ can then be obtained:

$$\mathbf{f}' | \mathbf{f}, \mathbf{X}, \mathbf{X}' \sim \mathcal{N}(\mu_C, K_C) \ , \tag{13}$$

$$\text{with } \mu_C = m(\mathbf{X}') + K(\mathbf{X},\mathbf{X}')^T K(\mathbf{X},\mathbf{X})^{-1}(\mathbf{y} - m(\mathbf{X})) \ , \tag{14}$$

and

$$\text{with } K_C = K(\mathbf{X}',\mathbf{X}') - K(\mathbf{X},\mathbf{X}')^T K(\mathbf{X},\mathbf{X})^{-1} K(\mathbf{X},\mathbf{X}') \ , \tag{15}$$

where $(K(\mathbf{X},\mathbf{X}'))_{i,j} = \left(K(\mathbf{X}',\mathbf{X})^T\right)_{i,j} = K(X_i, X'_j)$ for $1 \leq i \leq N$ and $1 \leq j \leq N'$.

This conditional distribution provides a way to predict the mean output from given input points $\mathbf{X}'$ as well as the covariance. The main interest of GPR for our application is its ability to quantify the uncertainty of the predictions.

For most situations, the available outputs often display noisy values such that observations are given by $\mathbf{y} = f(\mathbf{X}) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2_{obs}\mathcal{I}_N)$, where $\mathcal{I}_N$ refers to the identity matrix. The noise is assumed Gaussian and independent identically distributed. The covariance function is modified by adding a white noise kernel,

$$k_{noise}(X_i, X_j) = k(X_i, X_j) + \sigma^2_{noise}\delta_{i,j} \quad \text{for } i,j \leq N \ , \tag{16}$$

where δ is the Kronecker symbol. Equation (13) holds if $K(\mathbf{X},\mathbf{X})$ is replaced by $K(\mathbf{X},\mathbf{X}) + \sigma^2_{noise}\mathcal{I}_N$. Similarly, $\sigma^2_{noise}\mathcal{I}_{N'}$ is added to $K(\mathbf{X}',\mathbf{X}')$ to predict the measurement noise.

Predictions with Gaussian processes require the inverse of $K_\sigma = K(\mathbf{X},\mathbf{X}) + \sigma^2_{noise}\mathcal{I}_N$. This inverse is obtained by a Cholesky decomposition that has good numerical stability properties. However, the matrix inversion has a complexity $O(N^3)$. This is why GPR does not scale well with very large data sets. In this case, the data set size is around 1000, which is sufficiently low to have acceptable training times. Exact GPR can be carried out.

III.A.3. Choice of Covariance Functions

The covariance function defines the regularity of the functions sampled from the Gaussian process. Different families of covariance functions exist and are used depending on the expected shape of the function to be learned. In this work, the radial basis function (RBF) (or squared-exponential) kernels and Matérn covariance kernels are used.

Definition III.2

Let x and $x' \in \mathbb{R}^I$. The anisotropic squared-exponential covariance function k_{SE} is a covariance function defined by

$$k_{SE}(x,x') = \sigma^2 \prod_{p=1}^{I} exp\left(-\frac{(x_p - x_p')^2}{2l_p^2} \right) , \tag{17}$$

where the parameters l_p are the correlation lengths for each input dimension p. The kernel is said to be isotropic if l_p is independent of p.

Squared-exponential covariance functions are infinitely differentiable. As a result, the corresponding Gaussian processes produce very smooth functions. The realizations of a Gaussian process with squared-exponential covariance and mean zero are (almost surely) infinitely differentiable.

Definition III.3

Let x and $x' \in R^I$. Let $v \in R^+$. The Matérn class of anisotropic covariance functions $k_{Mat,v}$ is defined by

$$k_{Mat,v}(x,x') = \sigma^2 \frac{2^{1-v}}{\Gamma(v)} \prod_{p=1}^{I} \left(\frac{\sqrt{2v}|x_p - x_p'|}{l_p} \right)^v K_v \left(\frac{\sqrt{2v}|x_p - x_p'|}{l_p} \right) , \tag{18}$$

where K_v is the modified Bessel function and $\Gamma(v)$ is the gamma function.

The Matérn functions are a broad class of covariance functions parametrized by v, which defines the regularity of the covariance kernel. When $v \to +\infty$, the covariance function approaches the squared exponential

$$\lim k_{Mat,v} \underset{v\to+\infty}{=} k_{SE} . \tag{19}$$

The realizations of a Gaussian process with covariance function $k_{Mat,v}$ and mean zero are (almost surely) n-differentiable for $n < v$. The larger v, the smoother the Gaussian process.

The Matérn covariances can be expressed as a product of an exponential and a polynomial of order n for $v = n + 1/2$.

The Matérn and squared-exponential covariance functions are widely used in GPR. Since the functions to be learned are quite smooth here, Matérn 5/2 functions are chosen in the GPRs. This class of covariance functions has been shown to provide the best performance for GPR for this work. Matérn 3/2 and squared-exponential kernels were also tested, but provided lower performance.

III.A.4. Selection of Hyperparameters

In order to provide reasonable predictions for regression or classification problems, a Gaussian process has to be trained. The goal of the training phase is to choose the best parameters in the covariance kernels based on the training data.

The common practice for selecting the hyperparameters is to find the values that maximize the marginal likelihood $p(\mathbf{y}|\mathbf{X})$. The marginal likelihood refers to the probability of the observations $\mathbf{y}$ integrated over all the possible function values $\mathbf{f}$ drawn from the Gaussian process. It is defined by

$$p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|\mathbf{f},\mathbf{X}) p(\mathbf{f}|\mathbf{X}) d\mathbf{f} . \tag{20}$$

From previous equations, $\mathbf{f}|\mathbf{X} \sim \mathcal{N}(\mathbf{0}, K(\mathbf{X},\mathbf{X}))$ and $\mathbf{y}|\mathbf{f} \sim \mathcal{N}(\mathbf{f}, \sigma^2 \mathcal{I}_{\mathcal{N}})$. The log-marginal likelihood is thus given by

$$\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^T K_\sigma^{-1}\mathbf{y} - \frac{1}{2}\log|K_\sigma| - \frac{N}{2}\log(2\pi) , \tag{21}$$

with $K_\sigma = K(\mathbf{X},\mathbf{X}) + \sigma^2_{noise}\mathcal{I}_N$. The notation $|A|$, where A is a square matrix, refers to the determinant of the matrix A. The log-marginal likelihood is optimized using common optimization algorithms. In our case, the limited memory Broyden-Fletcher-Goldfarb-Shannon algorithm for bound constraint (also known as L-BFGS-B) is used.[9] The optimization algorithm is restarted 10 times with different initial values for the hyperparameters. The optimal set of hyperparameters chosen is the one that provides the highest log-marginal likelihood of the 10 iterations. With this approach, the risk of being stuck in a local optimum is reduced. Once the optimal set of hyperparameters is found, predictions can be made using Eq. (13).

III.B. Multi-Output Gaussian Processes

The framework described in the previous paragraphs assumes that the outputs are one dimensional. However, in this study the outputs to be predicted are vectors (R, Y_∞, X_∞) with dimension $D = 3$. For multi-output predictions, specific methods are required. Throughout the next paragraphs, the notations are extended for the multi-output case. Namely, the outputs for the training set and the test set are now in the matrix form $\mathbf{y} \in \mathbb{R}^{N\times D}$ and $\mathbf{y}' \in \mathbb{R}^{N'\times D}$.

III.B.1. Independent Scalar Gaussian Processes

A first trivial approach is to train one Gaussian process for each output dimension. The Gaussian processes are trained independently from one another. This method was used in a first approach in this work though it has some strong flaws. Independent training of the Gaussian processes means that the correlations between the outputs are not taken into account during the training phase; some information is lost.

More importantly, since the Gaussian processes are trained independently, the outputs must be assumed independent, which means that one can only predict the variances of each output, but not the full covariance matrix. The objective of our method is to provide improved mean predictions, but also the full covariance of the predicted outputs in order to include it into the inverse problem resolution.

Thus, the goal is to build a multi-output Gaussian process model able to provide nondiagonal covariance predictions. In this work specifically, the outputs are strongly correlated with one another, which makes this objective all the more important. Building a covariance kernel for a multidimensional Gaussian process is nontrivial since the covariance function has to remain positive definite. Several methods were investigated.

III.B.2. Linear Model of Coregionalization

In order to build a multi-output covariance kernel, one possible method is to start off with independent scalar Gaussian processes and mix them with a transition matrix. The matrix must be chosen to guarantee a positive definite covariance kernel.[10] With this approach, it is possible to correlate the output channels while maintaining a positive definite covariance kernel. A brief description of this method is presented in the next paragraphs.

Let us consider Q independent scalar Gaussian processes. For simplicity, only zero-mean Gaussian processes are considered:

$$u_q \sim \mathcal{GP}\left(0, k_q(x,x')\right) \text{ for } 1 \leq q \leq Q\,. \tag{22}$$

These are called latent Gaussian processes. Now let us consider a real mixing matrix $W \in R^{D\times Q}$. Let f_d be the output for channel d. It is obtained by the matrix product of W and the vector of latent Gaussian processes,

$$f_d(x) = \sum_{q=1}^{Q} w_{d,q} u_q(x)\,. \tag{23}$$

The covariance between two sets of inputs $\mathbf{X} \in \mathbb{R}^{N\times I}$ and $\mathbf{X}' \in \mathbb{R}^{N'\times I}$ for two channels d and d' can be calculated as

$$Cov[f_d(\mathbf{X}), f_{d'}(\mathbf{X}')] = \sum_{q=1}^{Q} w_{d,q} w_{d',q} k_q(\mathbf{X},\mathbf{X}')\,. \tag{24}$$

It is possible to flatten the multi-output vectors into a one-dimensional column,

$$\mathbf{f}(\mathbf{X}) = (f_1(X_1), ..., f_1(X_N), ..., f_D(X_1), ..., f_D(X_N))^T \in \mathbb{R}^{DN}\,. \tag{25}$$

Now the covariance matrix can be written as a $DN \times DN'$ matrix using the Kronecker product $\otimes$ as

$$\begin{aligned}\mathbf{K_{LMC}}(\mathbf{X},\mathbf{X}') &= Cov[\mathbf{f}(\mathbf{X}), \mathbf{f}(\mathbf{X}')] \\ &= \sum_{q=1}^{Q} W_{\cdot,q} W_{\cdot,q}^T \otimes \mathbf{K_q}(\mathbf{X},\mathbf{X}')\,,\end{aligned} \tag{26}$$

where $W_{\cdot,q}$ is the q'th column of the matrix W and $\mathbf{K_q}(\mathbf{X},\mathbf{X}')$ is the covariance matrix obtained for kernel k_q applied to the input sets $\mathbf{X}$ and $\mathbf{X}'$. This defines a covariance kernel for the multi-output Gaussian process, where

$$\left(\mathbf{K_{LMC}}(\mathbf{X},\mathbf{X}')\right)_{((d-1)N+i,(d'-1)N'+j} = Cov[f_d(X_i), f_{d'} \quad (X'_j)]$$

for $d, d' \leq D$, $i \leq N$ and $j \leq N'$.

The matrix inversion now has a complexity of $O\left(D^3N^3\right)$. Thus, the multi-output Gaussian processes are much more costly due to the size of the covariance matrix. For a large output dimension, approximation methods such as the sparse variational Gaussian process could be used.[11] In this work, the complexity remains reasonable and exact GPR can be carried out. With this method, it is possible to build a multi-output Gaussian process able to provide the full covariance matrix of the predictions.

III.B.3. Convolutional Gaussian Processes

Similarly, one can build convolutional Gaussian processes by mixing the latent Gaussian processes with a convolution product instead of a matrix multiplication.[12] The output for channel d is built by the following relation:

$$f_d(x) = \sum_{q=1}^{Q} \int G_{d,q}(x-z) u_q(z) dz\,, \tag{27}$$

where the functions $G_{d,q}$ represent filters in the convolution kernel.

Let $G_{d,q}$ be a Gaussian filter with covariance P_d^{-1}, a positive definite matrix. Then the density $G_{d,q}(x)$ is the density of a Gaussian random variable $\mathcal{N}(0, P_d^{-1})$ with a multiplicative constant $S_{d,q}$.

Let us consider the inputs $\mathbf{X} = (X_i)_{i\leq N} \in \mathbb{R}^{N\times I}$ and $\mathbf{X}' = (X'_j)_{j\leq N'} \in \mathbb{R}^{N'\times I}$. Let k_q be the squared-exponential covariance function corresponding to the q-th latent Gaussian process, with $q \leq Q$. It is written in a matrix form with $\Lambda_q = \text{diag}(l_{p,q}^{-2})_{p\leq I} \in \mathbb{R}^{I\times I}$, a diagonal matrix containing the squared inverse correlation lengths $l_{p,q}^{-2}$ of the q-th latent Gaussian process for each input dimension $p \leq I$. The kernel k_q also has a variance parameter V_q. Let

us look at the covariance between X_i for $i \leq N$ and X'_j for $j \leq N'$:

$$k_q(X_i, X'_j) = V_q exp\left(-\frac{1}{2}(X_i - X'_j)^T \Lambda_q (X_i - X'_j)\right). \tag{28}$$

The convolution product in the expression of the covariance becomes tractable,

$$Cov(f_d(X_i), f_{d'}(X'_j)) = \sum_{q=1}^{Q} \frac{S_{d,q} S_{d'}, q V_q}{(2\pi)^{I/2} |C_{d,d'}, q|^{1/2}} exp\left(-\frac{1}{2}(X_i - X'_j)^T C_{d,d',q}^{-1}(X_i - X'_j)\right), \tag{29}$$

with $C_{d,d',q} = P_d^{-1} + P_{d'}^{-1} + \Lambda_q^{-1}$. It is possible to define the full covariance matrix as

$$\mathbf{K_{CONV}}(\mathbf{X}, \mathbf{X'}) = \sum_{\mathbf{q=1}}^{\mathbf{Q}} \mathbf{V_q} \begin{pmatrix} \mathbf{S_{1,q}} \mathbf{S_{1,q}} \mathbf{K_{1,1,q}}(\mathbf{X}, \mathbf{X'}) & \dots & \mathbf{S_{D,q}} \mathbf{S_{1,q}} \mathbf{K_{1,D,q}}(\mathbf{X}, \mathbf{X'}) \\ \vdots & \ddots & \vdots \\ \mathbf{S_{D,q}} \mathbf{S_{1,q}} \mathbf{K_{D,1,q}}(\mathbf{X}, \mathbf{X'}) & \dots & \mathbf{S_{D,q}} \mathbf{S_{D,q}} \mathbf{K_{D,D,q}}(\mathbf{X}, \mathbf{X'}) \end{pmatrix}, \tag{30}$$

where $\mathbf{K_{d,d',q}}(\mathbf{X}, \mathbf{X'}) = \left(\mathbf{K_{d,d',q}}(X_i, X'_j)\right)_{i \leq N, j \leq N'}$ is given by

$$\mathbf{K_{d,d',q}}(\mathbf{X_i}, \mathbf{X'_j}) = (\mathbf{2\pi})^{-\mathbf{I/2}} |\mathbf{C_{d,d',q}}|^{-\mathbf{1/2}} \mathbf{exp}\left(-\frac{\mathbf{1}}{\mathbf{2}}(\mathbf{X_i} - \mathbf{X'_j})^{\mathbf{T}} \mathbf{C_{d,d',q}^{-1}}(\mathbf{X_i} - \mathbf{X'_j})\right). \tag{31}$$

On top of this, a white noise kernel is added for each of the output channels.

For convolutional Gaussian processes, the number of hyperparameters to optimize is much larger than for independent processes since the coefficients of the matrices P_d and Λ_q and the scalar $S_{d,q}$ have to be learned. To simplify the training process, the Gaussian filters are chosen with diagonal covariance matrices P_d. This reduces drastically the number of hyperparameters. The main flaw of these convolutional Gaussian processes is that they might provide functions that are too smooth due to the convolutional product with Gaussian filters and the use of RBF kernels.

III.C. Bias Learning

Instead of learning directly the Feynman moments and the average count rate, it is possible to learn the disparities between the point model and the simulated data. This idea is adapted from Ref. 13 where the low-fidelity code is the point model and the high-fidelity code is MCNP6 (or a real experiment):

$$y_{true} = f_{PM}(\theta) + f_{GP}(\theta) + \varepsilon, \tag{32}$$

where $f_{PM}(\theta)$ are the point model predictions for input θ and f_{GP} is the Gaussian process to be trained. Since the training data are not necessarily positive in this case, the Box-Cox transform cannot be used. Instead, the data are preprocessed using the Yeo-Johnson transform.[14]

However, the point model equations require neutron multiplicity values $(\overline{\nu}, D_2, D_3, \overline{\nu}_s, D_{2s}, D_{3s})$ for induced and spontaneous fissions. For predictions, one does not have information on the neutron multiplicity parameters. Two solutions could be considered. Either the nuclear multiplicity parameters are included in the inputs of the forward model, which increases the dimension of the input space to 13, or the nuclear multiplicity parameters are taken as the average over the training cases. However, this drastically limits the generalization of the surrogate model, especially if other fissile isotopes are considered. For simplicity purposes, the second approach is used here.

To account for systematic biases caused by the point model approximations, one can introduce a hyperparameter $\rho \in \mathrm{IR}$ and define a Gaussian process f_{GP} by weighting the contribution of the point model predictions with ρ,

$$y_{true} = \rho f_{PM}(\theta) + f_{GP}(\theta) + \varepsilon. \tag{33}$$

The training of the Gaussian process works similarly, except ρ is identified as an hyperparameter to be tuned by maximizing the marginal log-likelihood.

For the multi-output case, one can either take $\rho \in \mathrm{IR}$ as a scalar or $\rho \in \mathbb{R}^D$ as a vector and each of its component is weighting one output channel. The second approach is more flexible, but introduces more hyperparameters. It is preferred in this work since the additional computational cost is not prohibitive.

Overall, many different approaches have been presented for multi-output GPR. These methods are all implemented and their performances are evaluated in Sec. IV.C.2.

IV. GENERAL METHOD

IV.A. Extension of the Point Model

The first objective is to extend the point model using surrogate models based on Gaussian processes. For this purpose, the input space needs to be extended to include more input parameters and to describe more accurately the underlying physical processes.

First of all, the energy spectrum of the neutrons must be accounted for. Based on the two-group description commonly used in reactor physics, the spectral ratio Φ is introduced. It is defined as the ratio of thermal flux over fast flux, taken in the surrounding medium of the object studied:

$$\Phi = \frac{\phi_{th}}{\phi_{fast}} . \tag{34}$$

Next, parasitic absorptions should be included in the model description. In the point model, only the fissile material is considered, while in practice, neutron capture occurs in the moderating material as well (if any) or in the reflectors. In order to account for this, a new input parameter ε_A is defined as the ratio of absorptions outside the fissile material over induced fissions:

$$\varepsilon_A = \frac{\text{parasitic absorptions}}{\text{induced fissions}} . \tag{35}$$

Finally, the leakage is monitored by the ratio of outward over inward neutron current at the outermost layer of the MCNP6 model. In the training cases, this current ratio is taken at the concrete walls surrounding the object,

$$J_{ratio} = \frac{J_{out}}{J_{in}} . \tag{36}$$

Overall, the input space is now seven-dimensional (7-D) with input parameters $\theta = \left(k_p, \varepsilon_F, S, x_s, \varepsilon_A, \Phi, J_{ratio}\right)$. The MCMC sampling is more difficult since the dimension is higher. However, the additional parameters will be constrained using a simplified MCNP6 geometry that can be run quickly.

IV.B. Creating the Data Set

Sufficient training data must be fed to the surrogate models to construct efficient forward models. The training data were created using analog three-dimensional (3-D) Monte Carlo simulations with the code MCNP6 (Ref. 15).

The unknown object to be identified was a spherical fissile medium containing ^{239}Pu and ^{240}Pu and an inner void region of unknown diameter. It is surrounded by borated polyethylene, which serves as a moderator and a parasitic absorber. The object was placed in air at atmospheric pressure. The neutron detector consisted of a cylindrical tube of ^{3}He with CO_2 acting as a quench gas and a cylindrical external shell of polyethylene used to slow down the incoming neutrons. The geometry was surrounded by a concrete layer modeling the walls of the room.

The size, density, and composition of the different regions were changed for each simulation. The fraction of boron in the borated polyethylene was changed for each case, and simulations were also performed without boron at all. The volumic source term was a mix of a spontaneous fission source and an (α, n) source located uniformly in the fissile region. The ratio of spontaneous fissions over (α, n) reactions was randomly changed for each training case in order to explore all the possible x_s.

The prompt multiplication factor k_p was obtained with a criticality calculation with 20 inactive cycles and 100 active cycles. The other parameters in the vector θ were obtained by tally measurements. The complete training set had a total of 1125 cases.

The intrinsic physics of fission processes is not described. The neutron multiplicity distribution was modeled by the Terrel distribution.[16]

For each simulation, the Feynman moments and the average count rate must be evaluated. All the neutron captures occurring in the detector were recorded in a time list file using the PTRAC command in MCNP6. This file was then postprocessed to extract the Feynman moments. Since only the duration between the beginning of the neutron history and the time of detection was recorded in MCNP6, the birth times of the source neutrons must be sampled in the postprocessing step. The source, whether it was a spontaneous fission source, an (α, n) source, or a mix of both, was assumed to follow Poisson statistics with intensity S. Then the birth instants of the source neutrons were defined by drawing samples from these Poisson statistics.

Based on the time list file obtained after the postprocessing step, two main methods were used to evaluate the Feynman moments. They are described in the next paragraphs.

IV.B.1. Sequential Binning

In sequential binning, the numerical experiment is split into W time windows of size T. In each window i,

the number of counts $n_i(T)$ is recorded. From this, the simple moments of the detection statistics M_p can be estimated,

$$\widehat{M_p} = \frac{1}{W}\sum_{i=1}^{W} n_i(T)^p \, . \tag{37}$$

In the derivation of the point model equations, it can be seen that the Feynman moments are directly linked to the simple moments by the following relations:

$$R = \frac{M_1}{T} \ , \tag{38}$$

$$Y = \frac{M_2}{M_1} - M_1 - 1 \ , \tag{39}$$

and

$$X = \frac{M_3}{M_1} + 2(M_1^2 + 1) - 3\left(\frac{M_2}{M_1} + M_2 - M_1\right) . \tag{40}$$

And thus, estimators of the Feynman moments can be obtained by replacing the simple moments by their estimators $\widehat{M_p}$ in the previous equations.

While the average count rate R is independent of the size of the time window T, the Feynman moments are not. Once they are obtained for a given T, the same protocol can be repeated while merging $n \geq 1$ windows together to obtain estimators of $Y(nT)$ and $X(nT)$.

Since the outputs of interest are the asymptotic Feynman moments Y_∞ and X_∞, T should be large enough to reach the asymptotic state. However, the choice of a larger T leads to less time windows W, and thus more noisy data due to accidental correlations. A compromise had to be made for the choice of T between asymptoticity and noise. A time gate width of $\mathrm{T}_\infty = 3$ ms was used in this work.

The sequential binning method can be used for numerical or practical experiments as long as a time list file is available. In this work, it was used to mimic the practical evaluation of Feynman moments. All simulated data $\mathbf{y}$ from which the posterior distribution $p(\theta|\mathbf{y})$ is inferred were thus obtained using sequential binning. However, when sequential binning is used to obtain training data for the surrogate models, the inherent noise in the simulated data caused by accidental correlations reduces the performance of the surrogate models. Thus, another method was used to evaluate the Feynman moments for that specific task.

IV.B.2. Filtered Triggered Binning

The filtered triggered binning method allows for filtering out accidental correlations using the knowledge provided by the numerical simulation. In this method, time windows are opened whenever a neutron is detected, and the history number of this neutron is kept in memory. For a given detection, the history number of the neutron is written in the time list file alongside the detection time. Then the detections are recorded in the window if and only if the history number of the detected neutron is the same as the history number of the neutron that triggered the opening of the window. This means that only the correlated detections are recorded. An illustration of the method is shown in Fig. 1.

For a given window i, the recorded number of counts is $n_{i,tr}(T)$. The average number of double and triple counts, which are coincidentally the second and third binomial moments of the detection distribution, can then be obtained by the following estimators:

$$\widehat{m_p} = \frac{1}{N_{dec}}\sum_{i=1}^{N_{dec}}\prod_{k=0}^{p-2}\left(n_{i,tr}(T) - k\right) \ , \tag{41}$$

where N_{dec} is the total number of detections, and thus the total number of triggered windows. The Feynman moments can then be evaluated by

$$\widehat{R} = \frac{N_{dec}}{T_{tot}} \ , \tag{42}$$

$$\widehat{Y} = 2\widehat{m_2} \ , \tag{43}$$

and

$$\widehat{X} = 6\widehat{m_3} \, . \tag{44}$$

Filtered triggered binning is able to filter out the noise in the data, but is only applicable because numerical simulations are used and the history number of the neutron is known. For a practical measurement, only sequential binning can be used. For this reason, filtered triggered binning was only used to create the training set for the surrogate models. Some inherent noise remained

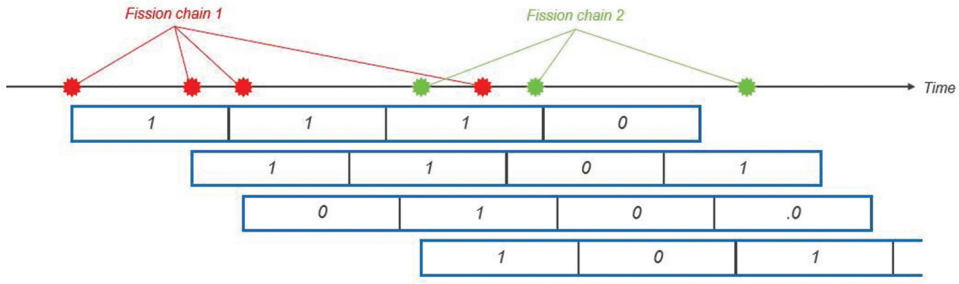

Fig. 1. Illustration of the filtered triggered binning method.

due to the stochastic nature of the MCNP6 simulations. These stochastic uncertainties were included in the Gaussian process covariance predictions.

Once the surrogate models were trained, the inverse problem was solved with data obtained by sequential binning in order to mimic the experimental data. Using filtered triggered binning data as training was a way to reduce the intrinsic noise in the data set and improve the training of the surrogate models.

IV.C. Training the Surrogate Models

IV.C.1. Data Set Preprocessing

Before training the surrogate models, the data set needed to be preprocessed to improve the training performance. With the noise introduced by accidental correlations, the Feynman moments obtained for low $k_p < 0.6$ can become negative. These irregular data were removed from the data set.

Then, since the data distribution is far from Gaussian, a Box-Cox transform was applied.[17] The Box-Cox transform is a nonlinear transformation whose goal is to reshape the distribution into a standard normal distribution $\mathcal{N}(0, \mathcal{I})$. It requires positive data,

$$B(y,\lambda) = \begin{cases} \frac{(y^{\lambda}-1)}{\lambda} & if\ \lambda \neq 0 \\ \log(y) & if\ \lambda = 0 \end{cases} . \tag{45}$$

The parameter λ is fitted to match a standard normal distribution by maximizing a log-likelihood between the data and the standard normal distribution.

IV.C.2. Gaussian Process Performance Comparison

Before training, the data set was split into a training set and a test set of the same size. The training set contained around 560 cases. Different metrics were used to evaluate the performance on the test set. The mean absolute error (MAE), mean squared error (MSE), and mean absolute percentage error (MAPE) are used here. Let $y_{d,j}$ be the j'th test output for the output channel d, corresponding to inputs θ_j for $j \leq N'$. Let $\overline{f_d(\theta_j)}$ be the mean surrogate model prediction for inputs θ_j and for the output channel d. Then for the output channel d the metrics are defined by

$$\mathrm{MAE}_d = \frac{1}{N'} \sum_{j=1}^{N'} \left| y_{d,j} - \overline{f_d(\theta_j)} \right| , \tag{46}$$

$$\mathrm{MSE}_d = \frac{1}{N'} \sum_{j=1}^{N'} \left(y_{d,j} - \overline{f_d(\theta_j)} \right)^2 , \tag{47}$$

and

$$\mathrm{MAPE}_d = 100 \times \frac{1}{N'} \sum_{j=1}^{N'} \left| \frac{y_{d,j} - \overline{f_d(\theta_j)}}{y_{d,j}} \right| . \tag{48}$$

The robustness of the uncertainty quantification was also investigated. The coverage probability in the estimated 2σ confidence interval is provided. It is defined as the ratio of test outputs lying inside the 2σ confidence interval predicted by the surrogate model. The theoretical value should be 95.45%.

In the first approach, the Gaussian processes were used to learn directly the observations (R, Y_∞, X_∞). The methods investigated were the multi-output independent Gaussian processes, the linear model of coregionalization (LMC), and the convolutional Gaussian processes described in Sec. III.B. Their performance is displayed in Table II.

The convolutional Gaussian processes did not provide satisfactory predictions. This is likely linked to the use of Gaussian filters and covariance kernels that hinder the Gaussian process in accurately modeling the data.

TABLE II

Performance of the Gaussian Process (GP) Surrogate Models on the Test Set

Independent GPs	MAE	MSE	MAPE	Coverage probability (2σ)
Count rate	1.98×10^{2}	1.10×10^{5}	0.95%	95.3%
Second Feynman	2.93×10^{-2}	3.84×10^{-3}	2.56%	94.2%
Third Feynman	1.25×10^{0}	4.71×10^{1}	9.67%	93.9%
LMC with two latent GPs	MAE	MSE	MAPE	Coverage probability (2σ)
Count rate	2.24×10^{2}	1.18×10^{5}	1.11%	92.2%
Second Feynman	3.77×10^{-2}	1.01×10^{-2}	2.85%	94.0%
Third Feynman	1.19×10^{0}	4.25×10^{1}	9.73%	94.6%
LMC with three latent GPs	MAE	MSE	MAPE	Coverage probability (2σ)
Count rate	2.20×10^{2}	1.09×10^{5}	1.10%	93.3%
Second Feynman	3.85×10^{-2}	9.22×10^{-3}	3.08%	90.5%
Third Feynman	1.33×10^{0}	5.96×10^{1}	10.0%	92.3%
Convolutional GPs	MAE	MSE	MAPE	Coverage probability (2σ)
Count rate	1.77×10^{3}	4.86×10^{7}	5.25%	98.4%
Second Feynman	4.62×10^{-1}	8.26×10^{0}	7.76%	97.9%
Third Feynman	4.10×10^{1}	1.34×10^{5}	18.3%	97.7%

Because of their poor performance, convolutional Gaussian processes are not studied in the next paragraphs.

On the other hand, both linear coregionalization Gaussian processes and independent Gaussian processes provided satisfactory mean predictions and one-dimensional (1-D) coverage probability. However, the independent Gaussian processes were expected to yield poor predictions for the two-dimensional (2-D) confidence ellipses. Thus, the coverage probability for the 2-D confidence ellipses were also investigated. Figure 2 displays the coverage probability for each couple of outputs and for the different surrogate models built. The coverage probability is shown for different levels of confidence.

From Fig. 2, one can see that using independent Gaussian processes for the output channels did not yield robust covariance predictions. Besides, convolutional Gaussian processes tended to overestimate the variance regardless of the confidence levels required. On the other hand, LMC provided robust covariance predictions with two or three latent Gaussian processes.

Overall, linear coregionalization Gaussian processes provided excellent predictions for the average count rate and the Feynman moments with robust covariance predictions. For this particular case, LMC with two latent Gaussian processes was the best-performing surrogate model, though its performance

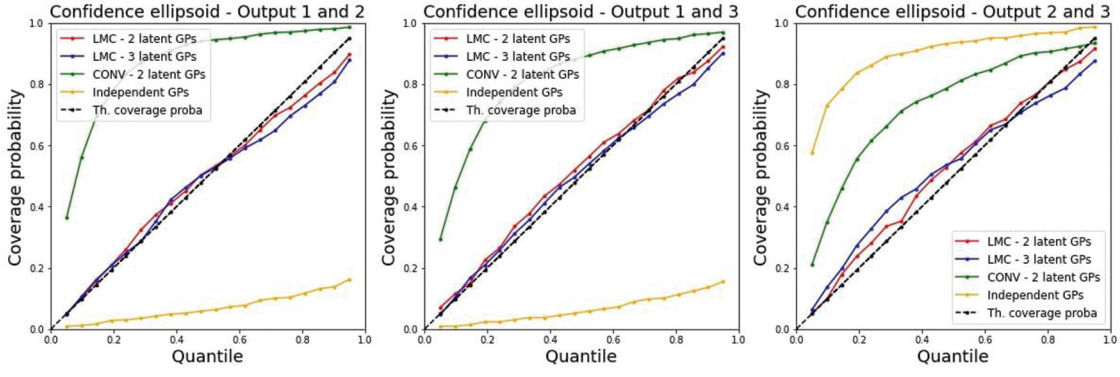

Fig. 2. Two-dimensional coverage probability for different levels of confidence obtained for the surrogate models: (red) LMC with two and (blue) three latent Gaussian processes, (green) convolutional Gaussian process, and (yellow) independent Gaussian processes.

was comparable to the LMC with three latent Gaussian processes.

In the second approach, the Gaussian processes learned the disparities with the point model, first with fixed bias $\rho = 1$ (Table III) and then with nonfixed ρ (Table IV). One can see that for all the surrogate models considered, the bias learning yielded overall better performance. Fixing $\rho = 1$ appears to be the best performing method. When ρ was not fixed, the optimization step was made more difficult due to the three additional parameters (for multi-output Gaussian processes), which can in turn impact the performance of the surrogate model.

Overall, this study showed that LMC with two or three latent Gaussian processes was able to provide robust surrogate models with good prediction performance. The surrogate models can be further improved by learning the bias with the point model instead of directly learning the outputs (R, Y_∞, X_∞). For the rest of this work, the surrogate model used is the LMC Gaussian process with two latent Gaussian processes.

IV.C.3. Sensitivity Analysis

Once the Gaussian processes were trained, a sensitivity analysis could be performed to evaluate the impact of the additional parameters $(\varepsilon_A, \Phi, J_{ratio})$. The objective was to understand whether or not the additional parameters affected the predictions.

The sensitivity analysis was done by evaluating the Sobol indices of total order $S_{T,p}$ for input p. Let $\mathbf{X}$ be a random variable of dimension I representing the inputs, and Y a scalar output random variable. The input coordinates are assumed independent. Let $\mathbf{X}_{-p} = (X_1, ..., X_{p-1}, X_{p+1}, ..., X_I)$ be the vector of input variables without the p'th coordinate. Then the total order Sobol index for dimension p is given by the relative fraction of the variance explained by the interaction of the p'th coordinate with all the other inputs:

$$S_{T,p} = \frac{\mathbb{E}\left[Var(Y|\mathbf{X}_{-p})\right]}{Var(Y)} . \tag{49}$$

The Sobol indices can be estimated with Monte Carlo estimators by the pick-freeze method.[18]

The total order Sobol indices are shown in Table V. As expected, the main input of importance was k_p for the Feynman moments and S played a significant role in the evaluation of the count rate R, while having very limited influence on the Feynman moments. One can observe that each of the additional inputs had an impact on at least one of the outputs that are similar to x_s. Thus, the additional inputs were indeed significant in the evaluation of the outputs, though their respective impacts were much more limited than for (k_p, ε_F, S).

IV.D. Adaptive MCMC

The sampling of the posterior distribution $p(\theta|\mathbf{y})$ of the parameter θ given simulated data $\mathbf{y}$ can be conducted directly with Bayes' theorem by choosing a fine mesh of the parameter space and by evaluating the likelihood in every point of the mesh. In the framework of the point model, the likelihood can be directly calculated, however this method is very cumbersome because it requires a fine discretization of the four-dimensional parameter

TABLE III

Performance of the Bias Learning Gaussian Processes (GPs) with $\rho = 1$ on the Test Set

Independent GPs	MAE	MSE	MAPE	Coverage probability (2σ)
Count rate	2.02×10^2	1.31×10^5	0.83%	95.7%
Second Feynman	5.88×10^{-2}	1.05×10^{-1}	2.78%	93.1%
Third Feynman	1.06×10^0	3.93×10^1	8.95%	94.2%
LMC with two latent GPs	MAE	MSE	MAPE	Coverage probability (2σ)
Count rate	2.05×10^2	1.14×10^5	0.94%	93.9%
Second Feynman	3.52×10^{-2}	7.93×10^{-3}	2.74%	93.9%
Third Feynman	1.08×10^0	3.42×10^1	8.51%	94.8%
LMC with three latent GPs	MAE	MSE	MAPE	Coverage probability (2σ)
Count rate	2.06×10^2	1.13×10^4	0.96%	93.1%
Second Feynman	3.30×10^{-2}	7.47×10^{-3}	2.54%	93.1%
Third Feynman	1.07×10^0	3.61×10^1	8.63%	96.0%

TABLE IV

Performance of the Bias Learning Gaussian Processes (GPs) with $\rho \neq 1$ on the Test Set

Independent GPs	MAE	MSE	MAPE	Coverage probability (2σ)	ρ
Count rate	2.00×10^{2}	1.17×10^{5}	0.93%	93.0%	1.038
Second Feynman	3.44×10^{-2}	8.23×10^{-3}	2.38%	95.8%	0.950
Third Feynman	9.23×10^{-1}	2.74×10^{1}	9.19%	95.7%	0.964
LMC with two latent GPs	MAE	MSE	MAPE	Coverage probability (2σ)	ρ
Count rate	2.08×10^{2}	1.27×10^{5}	0.96%	92.6%	1.022
Second Feynman	3.34×10^{-2}	6.83×10^{-3}	2.65%	94.2%	0.962
Third Feynman	1.12×10^{0}	4.09×10^{1}	8.44%	95.4%	0.923
LMC with three latent GPs	MAE	MSE	MAPE	Coverage probability (2σ)	ρ
Count rate	2.08×10^{2}	1.22×10^{5}	0.96%	92.2%	1.032
Second Feynman	3.83×10^{-2}	8.51×10^{-3}	3.11%	91.8%	0.941
Third Feynman	1.18×10^{0}	4.56×10^{1}	8.70%	94.0%	0.964

TABLE V

Total Order Sobol Indices Estimated with Pick-Freeze Method with 10 000 Monte Carlo Samples

Total Order Sobol	R	Y_{∞}	X_{∞}
k_p	3.72×10^{-1}	6.99×10^{-1}	8.16×10^{-1}
ε_F	6.44×10^{-1}	4.54×10^{-1}	5.21×10^{-1}
S	1.00×10^{-1}	1.48×10^{-4}	3.02×10^{-4}
x_s	4.30×10^{-3}	2.82×10^{-3}	1.39×10^{-3}
ε_A	5.59×10^{-4}	5.25×10^{-4}	1.30×10^{-3}
Φ	2.29×10^{-5}	1.39×10^{-3}	3.22×10^{-3}
J_{ratio}	4.55×10^{-3}	2.00×10^{-4}	4.03×10^{-4}

space. It becomes even more difficult to implement as the number of dimensions increases. Working with the surrogate models instead of the point model would require the discretization of a 7-D space, which is very costly in terms of memory and running time.

An alternative way of sampling the posterior distribution is to use the MCMC methods. The MCMC methods allow for creating random samples whose empirical distributions converge toward the target distribution. Besides, they only require knowledge of the target density within a multiplicative constant.

Markov chains are stochastic processes where one state only depends on the previous state. A Markov chain is said to be ergodic for a distribution π if the expectation of a function with respect to this distribution (assuming it exists) can be approximated by the empirical average of this function on the states of the chain.

More specifically, let X be a random variable following a law π. Let $(X_i)_{i \in \mathbb{N}}$ be an ergodic Markov chain for the distribution π. Then for any function f such that $\mathbb{E}_{\pi}[|f|] = \int_{\mathcal{X}} |f(x)|\pi(dx) < +\infty$,

$$\mathbb{E}_{\pi}[f] = \lim_{N \to +\infty} \frac{1}{N} \sum_{i=1}^{N} f(X_i)\,. \tag{50}$$

The goal of the MCMC algorithms is to build an ergodic Markov chain for a given target distribution. For this work, the target distribution to sample was the posterior distribution of the input parameters given some observations $p(\theta|\mathbf{y})$. The ergodic property allowed for estimating different quantities of interest for the distribution, such as its mean, its variance, all the moments, the quantiles, the probabilities of being in a given interval, and so on.

IV.D.1. Adaptive Metropolis

One of the simplest MCMC algorithms is the Metropolis-Hastings algorithm.[19,20] Its goal is to create an ergodic Markov chain whose invariant distribution is the target distribution π.

The Metropolis-Hastings algorithm is robust, but has some limitations in the case of very degenerate target distributions as is the case in this work. A degenerate distribution is a distribution whose support lies mainly on a subspace (or a manifold) of the parameter space, whose dimension is strictly lower. It can be, for example, a curve or a plane in the 7-D parameter space. This definition of degeneracy is not rigorous. More precisely, a degenerate distribution is rigorously defined as a distribution whose support has a Lebesgue measure equal to zero. However, the notion of degeneracy is considered in this work as a practical limitation to MCMC methods and not as a formal mathematical definition.

In such a case, since the support of the distribution is thin, most of the candidate points in Metropolis-Hastings tend to miss the support and be rejected, and consequently, the acceptance rate is close to 0. If the proposal covariance is adjusted to reach a higher acceptance rate, the distribution is not properly sampled and the chain stays around the same spot. One way to correct this is to adapt the covariance matrix of the proposal distribution in order to draw candidate points closer to the support of the distribution. The Adaptive Metropolis (AM) algorithm presented in this section is more thoroughly detailed in Ref. 21.

Let π be the target distribution and θ_0 the initial point of the chain. In Metropolis-Hastings, at each iteration $n \geq 1$ a candidate point is sampled with a proposal distribution $\widehat{\theta_n} \sim q(\cdot|\theta_{n-1}) \sim \mathcal{N}(\theta_{n-1}, \mathcal{C})$, where the proposal covariance $\mathcal{C}$ is usually chosen diagonal. The idea of the AM algorithm is to adapt the covariance of the proposal by estimating the empirical covariance of the previously accepted points of the chain. In this method, the proposal distribution is of the form $\widehat{\theta_n} \sim \mathcal{N}(\theta_{n-1}, \mathcal{C}_{n-1})$, but the covariance is modified at each step to match the empirical covariance of the points of the chain,

$$\begin{aligned} \mathcal{C}_n &= s \times Cov(\theta_0, ..., \theta_n) \text{ with } Cov\ (\theta_0, ..., \theta_n) \\ &= \frac{1}{n} \sum_{i=0}^{n} \left(\theta_i - \overline{\theta_n}\right) \left(\theta_i - \overline{\theta_n}\right)^T , \end{aligned} \tag{51}$$

with $\overline{\theta_n} = \frac{1}{n+1} \sum_{i=0}^{n} \theta_i$. The scalar s is a scaling parameter that needs to be tuned to reach the desired acceptance rate.

The direct calculation of the empirical covariance is cumbersome when the chain becomes long. A recursive formula is preferred to evaluate $\mathcal{C}_n$ and $\overline{\theta_n}$:

$$\overline{\theta_{n+1}} = \frac{1}{n+2} \left(\theta_{n+1} + (n+1)\overline{\theta_n}\right) \tag{52}$$

and

$$\begin{aligned} \mathcal{C}_{n+1} = &\frac{n-1}{n}\ \mathcal{C}_n \\ &+ \frac{s}{n} \left(n\ \overline{\theta_n}\ \overline{\theta_n}^T - (n+1)\ \overline{\theta_{n+1}}\ \overline{\theta_{n+1}}^T + \theta_{n+1}\ \theta_{n+1}^T \right) . \end{aligned} \tag{53}$$

These recursive formulas help speed up the calculation of the covariance. In practice, it is advised to add a small term of the form $\epsilon \mathcal{I}_d$ with $\epsilon > 0$ in order to guarantee the matrix stays positive definite. Indeed, numerical approximations can lead to a degenerate covariance matrix, which can be problematic for the sampling of the candidate points.

The acceptance rate of the candidate points in the MCMC methods is a key factor to monitor. For Metropolis-Hastings, it was shown that the optimal acceptance rate was roughly 0.234 in high dimension.[22] The scaling factor of the proposal covariance must be tuned in order to reach an acceptance rate close to this value. A naïve search for a good scaling factor can be performed, but it is also possible to dynamically change the scaling factor to reach a target acceptance rate. One solution explored in Ref. 21 is to multiply the covariance at each step by a factor r_n defined as

$$r_n = \exp\left(\alpha_n - \alpha_{target}\right) , \tag{54}$$

where α_{target} is the target acceptance rate and α_n is the current acceptance rate. This method is implemented in the AM algorithm in our work.

The adaptation of the covariance matrix is not started directly from the beginning, but rather after a certain number of accepted points n_0 is reached in order to make sure the empirical covariance is calculated on enough points. n_0 was set at 500 in our case.

The AM algorithm is shown in Algorithm 1. It can be shown that the AM algorithm retains the ergodic property.[23]

The AM algorithm is more suited to degenerate probability distributions as the proposal distribution aligns with the distribution support and the candidate points are closer to the target distribution.

Algorithm 1: Adaptive Metropolis

Result: Sampling of the target distribution π
Choose the chain starting point θ_0 and desired length K;
Set the start of adaptation n_0
while *Chain length* $< K$ **do**
  Generate candidate $\widehat{\theta}$ from proposal distribution $q\left(\widehat{\theta}|\theta_n\right)$;
  Evaluate the acceptance probability $\alpha\left(\hat{\theta}, \theta_n\right) = min\left\{1 \frac{\pi\left(\hat{\theta}\right) q\left(\theta_n|\hat{\theta}\right)}{\pi\left(\theta_n\right) q\left(\hat{\theta}|\theta_n\right)}\right\}$;
  Generate $u \sim \mathcal{U}[0, 1]$ from a uniform distribution on [0, 1];
  if $\alpha\left(\widehat{\theta}, \theta_n\right) > u$ **then**
    Add $\widehat{\theta}$ to the chain $\theta_{n+1} = \widehat{\theta}$
    if *Number of iterations* $\geq n_0$ **then**
      Adapt the proposal covariance with equation 53;
    else
      Keep the same proposal covariance
    end
  else
    Add θ_n to the chain $\theta_{n+1} = \theta_n$;
  end
end

A variant of this algorithm is the Adaptive Proposal[24] where the covariance adaptation is performed locally using the H previous points instead of all the previous points. However, for this method the invariant measure is biased with respect to the target distribution. For this reason, the AM algorithm was used.

IV.D.2. Coupling MCMC and Surrogate Model Predictions

The forward model used to predict the outputs y for given inputs θ is found in the evaluation of the posterior density $p(\theta|\mathbf{y})$ in the AM algorithm when evaluating the acceptance probability $\alpha(\widehat{\theta}, \theta_n)$.

Let us consider N independent observations, with each including the count rate and the second and third Feynman moments $\mathbf{y} = (y_i)_{i \leq N} = ((R, Y_\infty, X_\infty)_i)_{i \leq N}$. Using Bayes' theorem, the posterior can be written as the product of a prior distribution $p(\theta)$ and a likelihood $L(\mathbf{y}|\theta)$, which is the probability distribution of the observations $\mathbf{y}$ given the inputs θ:

$$p(\theta|\mathbf{y}) \propto p(\theta) L(\mathbf{y}|\theta) \ . \tag{55}$$

It is assumed the N independent observations are given by $y = f(\theta) + \varepsilon$, where f is a forward model used to predict the outputs and $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_{obs})$. The forward model can be, for example, the analytical point model or a Gaussian process surrogate model. Then the acceptance probability in MCMC can be easily evaluated since the likelihood is Gaussian and the observations $\mathbf{y} = (y_i)_{1 \leq i \leq N}$ are independent,

$$L(\mathbf{y}|\theta) \propto \exp\left(-\frac{1}{2} \sum_{i=1}^{N} (y_i - f(\theta))^T \mathbf{C}_{obs}^{-1} (y_i - f(\theta))\right) . \tag{56}$$

This requires knowing the covariance of the noise $\mathbf{C}_{obs}$. This covariance is not simply a diagonal matrix because the three output channels are strongly correlated. The covariance could be estimated simply by taking the empirical covariance of the observations; however, this is not very efficient because the number of independent observations is typically around 10.

Instead, a bootstrap method was used to evaluate this covariance matrix.[25] The number of bootstrap samples

was set to 10 000. The outputs were strongly correlated as expected with $Corr(Y_\infty, X_\infty) \simeq 0.9$, for example. Hence the target density was known within a multiplicative constant. The MCMC sampling can now be performed using either the point model or a surrogate model for better predictions.

However, even though surrogate models based on GPR do perform better than the point model for predicting the outputs, they also come with uncertainties in their predictions. In order to have a robust uncertainty quantification method, these model uncertainties must be accounted for in the posterior distribution sampling.

In order to include the model uncertainties, the likelihood is modified. It is assumed that the model errors and the noise errors of the data are independent. Thus, the statistical model for a Gaussian process surrogate model can be rewritten as $y = \overline{f(\theta)} + \varepsilon_{model}(\theta) + \varepsilon_{noise}$ where the noise is $\varepsilon_{noise} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_{obs})$ and the model error is $\varepsilon_{model}(\theta) \sim \mathcal{N}(\mathbf{0}, Cov[f(\theta)])$. $\overline{f(\theta)}$ is the mean prediction of the Gaussian process and $Cov[f(\theta)]$ is the covariance prediction of the Gaussian process at input point θ.

Since the model error and noise are assumed independent, the likelihood can be modified to include the model error.[26] However, the simulated data are not independent anymore since they are all linked by the same model error $\varepsilon_{model}(\theta)$.

The residuals $\mathbf{R}(\theta) = \left(y_i - \overline{f(\theta)}\right)_{i \le N} \in R^{DN}$ and the total covariance $\mathbf{C}_{tot}(\theta) \in \mathbb{R}^{DN \times DN}$ are introduced,

$$\mathbf{C}_{tot}(\theta) = \begin{pmatrix} \mathbf{C}_{obs} + Cov[f(\theta)] & \dots & Cov[f(\theta)] \\ \vdots & \ddots & \vdots \\ Cov[f(\theta)] & \dots & \mathbf{C}_{obs} + Cov[f(\theta)] \end{pmatrix}. \tag{57}$$

The model error can be included in the MCMC sampling by the mean of a modified likelihood,

$$L(\mathbf{y}|\theta) \propto \frac{1}{\sqrt{|\mathbf{C}_{tot}(\theta)|}} \exp\left(-\frac{1}{2} \mathbf{R}(\theta)^T \mathbf{C}_{tot}(\theta)^{-1} \mathbf{R}(\theta) \right). \tag{58}$$

The evaluation of the likelihood is more cumbersome now since a matrix inversion and a determinant calculation must be performed at each iteration in MCMC, yet this is still manageable for low output dimensions. This new likelihood does include the model uncertainty in the sampling of the posterior distribution $p(\theta|\mathbf{y})$. It is used with the AM algorithm to sample the posterior distribution. In the next section, this method is used on a test case from the International Criticality Safety Benchmark Evaluation Project (ICSBEP) Handbook.

V. APPLICATION TO THE BERYLLIUM-REFLECTED PLUTONIUM SPHERE ICSBEP BENCHMARKS

The method presented in this paper was tested on two examples. The first one was taken from the ICSBEP Handbook[27] and the second one was based on the same fissile object with a different moderating material. Simplified descriptions of the two cases were built into MCNP6 and used to simulate neutron correlation data. From this, the posterior distribution of the parameters θ was sampled and compared to the MCNP6 values.

V.A. MCNP6 Modeling

V.A.1. Copper-Reflected Plutonium Sphere

The first example was taken from the set of experiments FUND-NCERC-PU-HE3-MULT-003 of the ICSBEP Handbook, experiment no. 1. The experiment is a measurement of neutron correlations on the beryllium-reflected plutonium (BERP) sphere, a metallic Pu sphere with a mean diameter of 7.5876 cm and an average density of 19.6039 $g \cdot cm^{-3}$. The sphere is surrounded by a SS-304 cladding consisting of two hemispheres. For simplicity, the hemispheres are modeled as a single spherical shell of SS-304 with an inner and outer diameter of 7.65556 and 7.71652 cm, respectively. The sphere is surrounded by a single layer of a spherical shell of copper reflector with an inner and outer diameter of 7.7978 and 10.1600 cm, respectively.

An aluminum structure supports the BERP ball. However, to simplify the MCNP6 model, this structure was not included. The expected prompt multiplication factor evaluated should be slightly lower than for the practical experiment due to the absence of the reflections on the support structure.

In the experiment, two NOMAD detectors were placed on each side of the BERP ball. In the simplified model, only one detector was modeled. The detectors have no influence on each other, as shown in the benchmark. The NOMAD detectors consisted of a series of 15 ^{3}He tubes with active dimensions of 2.46 × 38.1 cm set in a polyethylene moderating block. This design was simplified in the model used in this work. The detectors were replaced by a single cylindrical ^{3}He

tube. The diameter of the tube was chosen to have the same active volume as for the real experiment. The diameter of the tube was thus 9.52 cm. Similarly, the thickness of the polyethylene was set to 5.00 cm. This was chosen to have the same total polyethylene mass as in the experiment.

The detector region was filled with a mixture of ^{3}He with 2 at. % of CO_2 acting as a quench gas, where at. % refers to the atomic percentage of an element. The pressure was set to 10.13 bars.

The source intensity was chosen in the postprocessing step for the sampling of the neutron birth instants. The source intensity was chosen to be $S = 132\ 582$ events·s^{-1} and with $x_s = 0.969$.

The detector efficiency, the second Feynman moment $Y(T)$, and the prompt multiplication estimated by the neutron correlation observations were also provided.[28] The time gate width was set to $T = 2048$ μs in the benchmark. These values were compared to the ones obtained with the simplified MCNP6 model built to create our own Feynman observations. The second Feynman, as well as k_p and ε_F, are shown in Table VI with their respective standard deviations.

Overall, the simplified MCNP6 model built was close to the benchmark model and the experiment. Some disparities arose due to the simplifications made. They can be linked to the removal of the aluminum support plate, for example. Our goal here was not necessarily to exactly retrieve the benchmark results but rather to show the improvements brought by the method presented in this paper with an application to a well-documented experiment.

V.A.2. Polyethylene-Reflected BERP Sphere

In the training set, the fissile material was surrounded by borated polyethylene, and thus, the copper-reflected example displayed some disparities with the actual training examples. These disparities might bring a bias in the posterior distribution of θ. For this reason, a second example was tested, this time with the BERP ball surrounded by polyethylene. This example was taken from Ref. 29. The methodology was similar.

The polyethylene shell had an internal and external diameter of 7.798 and 15.418 cm and a density of 0.95 g·cm^{-3}. The fissile object was unchanged, and thus, the source intensity was the same, $S = 132\ 582$ events·s^{-1}.

The benchmark does not provide the prompt multiplication factor or the detector efficiency. The values obtained with the MCNP6 model were $k_p = 0.8967 \pm 0.0004$ and $\varepsilon_F = 0.0118 \pm 0.0001$. The second Feynman moment for $T = 2048$ μs is $Y_{exp}(T) = 0.75 \pm 0.05$, which is close to the value obtained with the simplified model used $Y_{MCNP}(T) = 0.814 \pm 0.014$.

As in the previous case, small disparities were observed between the experiment and the simplified MCNP6 model used in this work.

V.B. Posterior Distribution Sampling

The two cases described previously were used to create neutron correlation observations from which the posterior distribution $p(\theta|\mathbf{y})$ was sampled with the AM algorithm.

V.B.1. Sampling with the Point Model

The first approach was to use only the point model as a forward model. The likelihood is then given by Eq. (56). This method does not provide a way to quantify the bias of the point model. Only the observation covariance $\mathbf{C}_{obs}$ is added into the MCMC sampling.

The data were simulated by running 10 independent MCNP6 simulations, with 2×10^6 neutron histories for each. The Feynman moments were evaluated using the sequential binning method described in Sec. IV.B.1. The time gate width was set to $T_\infty = 3$ ms.

The AM algorithm was used to sample the posterior distribution. The sampling was performed over 5×10^6 iterations parallelized on five CPUs. The total running time of MCMC was 5 min. The first 2000 iterations were not recorded in the chain in order to make sure the chain had reached its stationary distribution. This is called the burn-in phase. The target acceptance rate was set to 0.12, which is lower than the usual 0.234 because the target distribution was very degenerate and far from a regular

TABLE VI

Comparison Between the ICSBEP Experiment and the Simplified MCNP6 Model for the Copper-Reflected BERP Ball

	$Y(T)$	k_p	ε_F
Experiment	0.339 ± 0.008	0.8306 ± 0.0009	0.0216
Simplified MCNP6	0.39 ± 0.01	0.8279 ± 0.002	0.0182 ± 0.0002

Gaussian target. The covariance adaption was started after 5000 iterations to ensure enough points were recorded in the chain so that the empirical covariance was not too degenerate. A regularization term was added to the proposal covariance to avoid reaching non-positive definite matrix because of numerical instabilities.

The prior $p(\theta)$ for $\theta = (k_p, \varepsilon_F, S, x_s)$ was set to a uniform distribution on a broad domain $\mathcal{D}$,

$$\mathcal{D} = [0.70, 0.95] \times [0.001, 0.100] \times [1 \times 10^5, 2 \times 10^5] \times [0.4, 1.0] . \quad (59)$$

The choice of the prior should be noninformative. This prior does not assume good prior knowledge of the input parameters. An even broader domain could be used, but this would require more training cases to completely cover the prior domain. Since exact GPR is used in this work, the prior domain is kept small enough even though this means the prior is not completely noninformative.

One could also use Jeffreys prior,[30] which is a noninformative prior designed to be invariant under reparameterization of the forward model. In the point model framework, Jeffreys prior is analytically tractable. However, it has been shown in a previous work that the effect of the prior for the MCMC sampling does not affect significantly the posterior distribution as long as enough observations are provided.[31]

The maximum a posteriori θ_{map} obtained was not really close to the real values:

$$\theta_{cu} = (0.856, 0.0128, 151500, 0.68) . \quad (60)$$

The 2-D marginal densities for (k_p, ε_F, S) are shown in Fig. 3. One can see that the real values of the parameters lie far outside the actual posterior distribution. This is a consequence of the bias of the point model that is not accounted for in this method.

The same calculations were performed for the polyethylene case. The 2-D marginal densities are displayed in Fig. 4. The maximum a posteriori θ_{poly} itself does not provide good predictions once again:

$$\theta_{poly} = (0.860, 0.0197, 108300, 0.83) . \quad (61)$$

For the polyethylene case, the real values lie closer to the sampled posterior distribution. Indeed, with the polyethylene, the neutrons were thermalized such that the assumption of monoenergetic neutrons is more reasonable than for the copper-reflected case.

Overall, the posterior distribution sampling with only the point model as a forward model did not provide robust uncertainty quantification nor a reliable prediction on the input parameters θ. This can be improved by using the surrogate models.

V.B.2. Prior Constraints on the Domain

In order to improve the sampling of the posterior distribution, the surrogate model was coupled to the MCMC sampling, as described in Sec. IV.D.2.

The input parameter space is now 7-D, which makes the MCMC sampling much more difficult. To counterbalance the higher dimension, more restrictive bounds were placed on the prior.

The parameters of interest are mainly $(k_p, \varepsilon_F, S, x_s)$. The three additional parameters have less impact on the predictions. They can be constrained either by direct practical measurement or by numerical simulations with MCNP6. In this work, a few 1-D MCNP6 calculations were run to estimate these parameters. For each simulation, the parameters

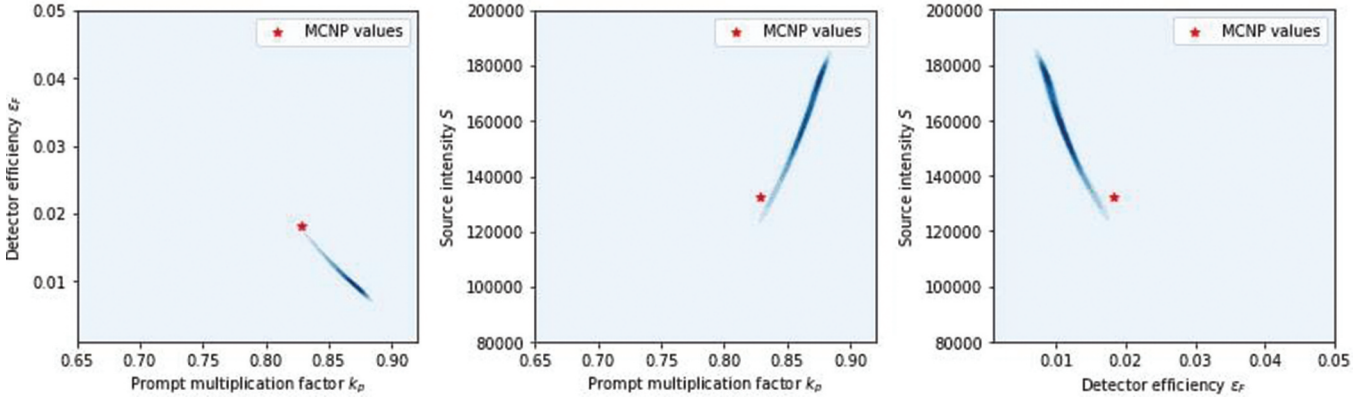

Fig. 3. Two-dimensional marginal densities of the posterior distribution sampled by MCMC with only the point model for the copper-reflected sphere.

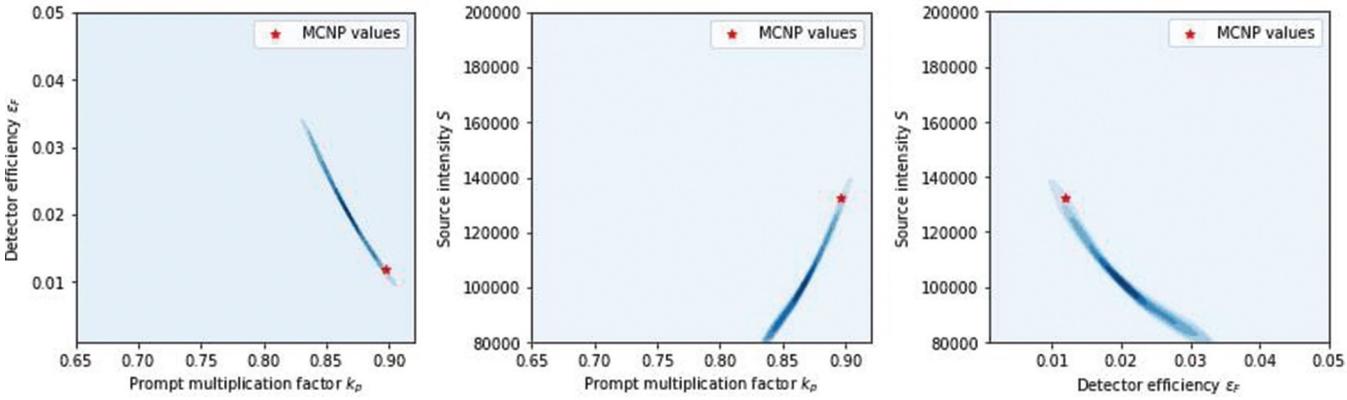

Fig. 4. Two-dimensional marginal densities of the posterior distribution sampled by MCMC with only the point model for the polyethylene-reflected sphere.

were evaluated by tally measurements. It was assumed the external diameter of the fissile region was known and fixed. This information can be obtained by γ spectroscopy, for example. The internal diameter was changed for each simulation.

The sizes of the 1-D spherical regions were chosen to preserve the total mass compared to the 3-D model. Since the detector was all around the fissile object, a correction based on the real solid angle Ω of the detector as seen by the fissile region was applied. Only $\Omega/4\pi$ neutrons were actually recorded in the ^{3}He detector.

The bounds for the uniform prior for $(\varepsilon_A, \Phi, J_{ratio})$ were then chosen as the minimal and maximal values obtained in the 1-D simulations. The prior $p(\theta)$ was a uniform distribution on the extended domain $\mathcal{D}_{ext}$,

$$\mathcal{D}_{ext} = \mathcal{D} \times [3 \times 10^{-3}, 5 \times 10^{-3}] \times [10^{-4}, 10^{-3}] \times [1.7, 1.9]\,. \quad (62)$$

V.B.3 Surrogate Model Sampling

The surrogate model used was the bias learning linear coregionalization Gaussian process with two latent processes and nonfixed $\rho \neq 1$. This surrogate model displayed the best performance, though it was comparable to the other LMC surrogate models. The posterior distribution sampling was performed similarly as in Sec. V.B.1. The MCMC sampling was done over 5×10^6 iterations once again, with a total running time of 40 min. The running time was slightly longer since the modified likelihood evaluation in Eq. (58) was more computationally demanding.

The marginal densities for the copper- and polyethylene-reflected Pu sphere are displayed, respectively, in Figs. 5 and 6. The maxima a posteriori for the copper θ_{cu} and the polyethylene θ_{poly} are also given as

$$\theta_{\text{cu}} = (0.806, 0.0222, 136000, 0.617) \quad (63)$$

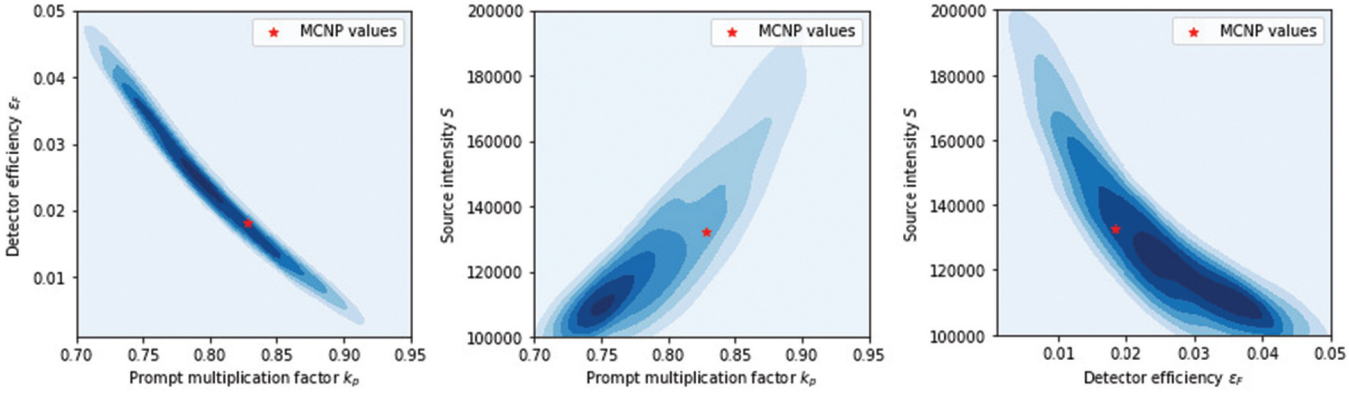

Fig. 5. Two-dimensional marginal densities of the posterior distribution sampled with the bias learning LMC2 surrogate model for the copper-reflected case.

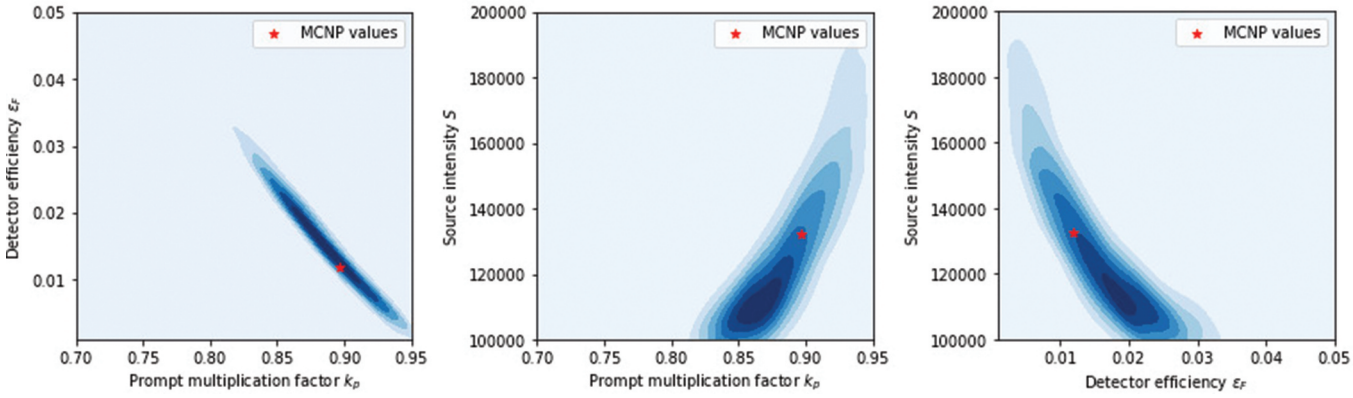

Fig. 6. Two-dimensional marginal densities of the posterior distribution sampled with the bias learning LMC2 surrogate model for the polyethylene-reflected case.

and

$$\theta_{poly} = (0.865, 0.0228, 118300, 0.74)\,. \tag{64}$$

The posterior distributions sampled were much broader because the model error was now included. For both cases, the theoretical points were in the distribution support. The maxima a posteriori were slightly more precise than in Sec. V.B.1, but do not provide precise predictions on their own. The estimation of x_s was especially difficult.

The training cases used a polyethylene moderator around the sphere, which means that the copper-reflected case differed from the training set used. Yet the real values were within the support of the distribution. The surrogate models were flexible enough to provide reasonable predictions on cases different from the training set.

The methodology presented in this paper allows for a robust inverse uncertainty quantification while being flexible in its use. As remarked previously, the posterior distributions were quite broad due to the significant model errors in the surrogate models (of course, another reason is the rather limited amount of information given by the neutron correlation observations considered in the test cases). The global performance of the method for a given data set could be improved by reducing the model error. This could be achieved by using more involved surrogate modeling approaches as we discuss in the next section. This would allow for obtaining narrower posterior distributions for real-world applications in nuclear safeguards or waste identification, for example.

V.C. Further Improvements

The surrogate models described in this paper are based on homoscedastic Gaussian processes. The white noise kernel added to account for the noise in the training data was assumed constant over the whole range of data. However, from the observations made in this work, the noise increased with k_p.

In Fig. 7, the log-residuals (in absolute value) of the GPR are plotted as a function of k_p for the test case. The residuals appear to be larger for large k_p. Figure 7 illustrates the heteroscedastic nature of the data used in this work.

In order to improve this methodology, heteroscedastic Gaussian processes could be created. For example, one can assume the variance in the training follows a parametric trend $\sigma^2_{noise}(x) = f_\beta(x)$. Then the parameters β can be included in the hyperparameters selection described in Sec. III.A.4. Other methods use a second Gaussian process as a surrogate model for the noise variance itself.[32]

The methodology is also compatible with other types of surrogate models as long as they are able to provide covariance predictions. Other supervised learning techniques could be applied to this problem. In this paper, the choice of the surrogate model was motivated by the simplicity and the flexibility of GPR, but more complex techniques could improve the posterior distribution sampling.

In this paper, the objective was not to model real-world measurements as accurately as possible, but rather to build a methodology able to solve the inverse problem and provide robust uncertainties on the predictions. In order to apply this methodology to real-world measurement, one would require better performance from the surrogate models. This could be achieved by adding inputs for the description of the energy spectrum or the slowing-down process. Besides, the surrogate models could benefit from a more detailed description of the fission dynamics, as is done with the fission models FREYA and FIFRELIN, for example.[33]

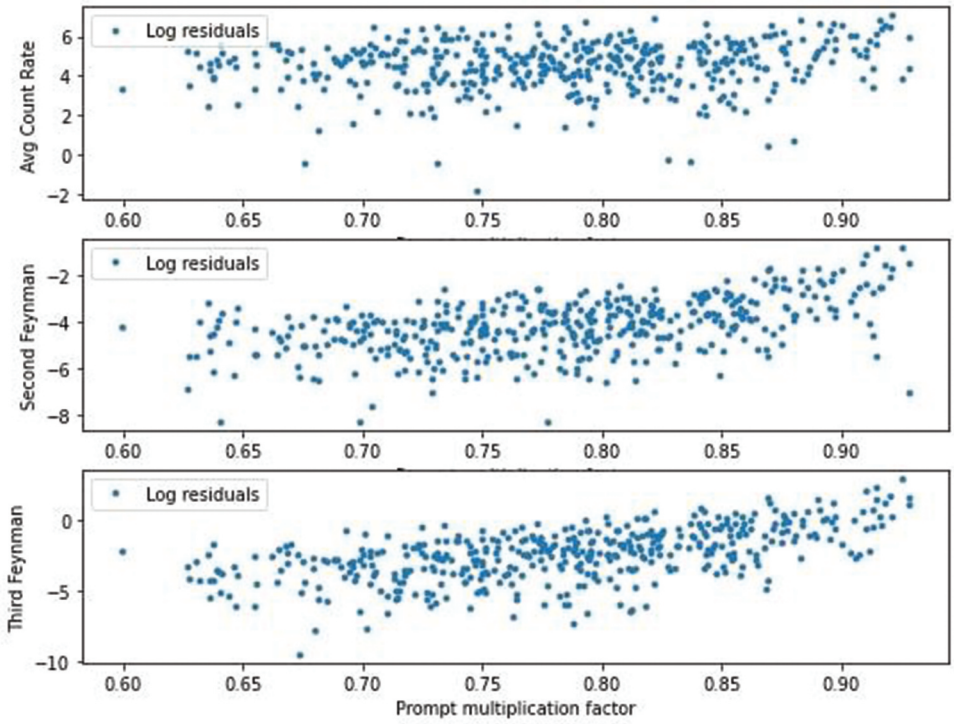

Fig. 7. Plot of the log residuals (in absolute value) for the test case as a function of k_p for the LMC2 GPR.

VI. CONCLUSION

Overall, the methodology presented in this paper provides a framework for robust uncertainty quantification in neutron noise analysis. The extension of the point model with the help of surrogate models based on Gaussian processes allows for better predictions in the forward model and more consistent uncertainty quantification in the inverse problem resolution.

The application to the two test cases highlighted the robustness of the method for uncertainty quantification. Yet, the trained surrogate models still suffered from large variances, which hindered the precise estimation of fissile mass based on neutron correlation observations. Two lines of work were identified to improve the practical application of this methodology to real-world problems in nuclear safeguards or waste drum identification. First of all, the surrogate models could be improved with the help of heteroscedastic Gaussian processes or other supervised learning techniques. On the other hand, one could also provide more information to the surrogate models to accurately model realistic neutron correlation measurements, such as better descriptions of slowing-down processes and fission dynamics.

Despite the possible improvements aforementioned, the methodology presented in this paper allows for robust uncertainty quantification for inverse problem resolution in nuclear safeguards, coupled with affordable computational resources, while improving the standard approach based on the analytical point model description.

Disclosure Statement

No potential conflict of interest was reported by the authors.

ORCID

Paul Lartaud http://orcid.org/0000-0001-6114-5762

References

1. R. P. FEYNMAN, F. DE HOFFMANN, and R. SERBER, "Dispersion of the Neutron Emission in U-235 Fission," *J. Nucl. Energy*, **3**, *1–2*, 64 (1956); https://doi.org/10.1016/0891-3919(56)90042-0.
2. I. PÁZSIT and L. PÁL, *Neutron Fluctuations: A Treatise on the Physics of Branching Processes*, Elsevier (2007).
3. D. M. CIFARELLI and W. HAGE, "Models for a Three-Parameter Analysis of Neutron Signal Correlation Measurements for Fissile Material Assay," *Nucl. Instrum.*

Methods Phys. Res., Sect. A, **251**, *3*, 550 (1986); https://doi.org/10.1016/0168-9002(86)90651-0.

4. J. M. VERBEKE, "Neutron Multiplicity Counting: Credible Regions for Reconstruction Parameters," *Nucl. Sci. Eng.*, **182**, *4*, 481 (2016); https://doi.org/10.13182/NSE15-35.
5. J. VERBEKE and O. PETIT, "Stochastic Analog Neutron Transport with TRIPOLI-4 and FREYA: Bayesian Uncertainty Quantification for Neutron Multiplicity Counting," *Nucl. Sci. Eng.*, **183**, *2*, 214 (2016); https://doi.org/10.13182/NSE15-82.
6. C. K. WILLIAMS and C. E. RASMUSSEN, *Gaussian Processes for Machine Learning*, Vol. 2, MIT Press, Cambridge, Massachusetts (2006).
7. K. BÖHNEL, "The Effect of Multiplication on the Quantitative Determination of Spontaneously Fissioning Isotopes by Neutron Correlation Analysis," *Nucl. Sci. Eng.*, **90**, *1*, 75 (1985); https://doi.org/10.13182/NSE85-2.
8. A. FURUHASHI and A. IZUMI, "Third Moment of the Number of Neutrons Detected in Short Time Intervals," *J. Nucl. Sci. Technol.*, **5**, *2*, 48 (1968); https://doi.org/10.1080/18811248.1968.9732402.
9. R. H. BYRD et al., "A Limited Memory Algorithm for Bound Constrained Optimization," *SIAM J. Sci. Comput.*, **16**, *5*, 1190 (1995); https://doi.org/10.1137/0916069.
10. E. V. BONILLA, K. CHAI, and C. WILLIAMS, "Multi-Task Gaussian Process Prediction," *Adv. Neural. Inf. Process Syst.*, **20**, 153 (2007).
11. M. TITSIAS, "Variational Learning of Inducing Variables in Sparse Gaussian Processes," *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, PMLR 5:567-574 (2009); https://proceedings.mlr.press/v5/titsias09a.html.
12. M. A. ALVAREZ and N. D. LAWRENCE, "Computationally Efficient Convolved Multiple Output Gaussian Processes," *J. Mach. Learn. Res.*, **12**, 1459 (2011).
13. M. C. KENNEDY and A. O'HAGAN, "Predicting the Output from a Complex Computer Code When Fast Approximations Are Available," *Biometrika*, **87**, *1*, 1 (2000); https://doi.org/10.1093/biomet/87.1.1.
14. I.-K. YEO and R. A. JOHNSON, "A New Family of Power Transformations to Improve Normality or Symmetry," *Biometrika*, **87**, *4*, 954 (2000); https://doi.org/10.1093/biomet/87.4.954.
15. T. GOORLEY et al., "Initial MCNP6 Release Overview," *Nucl. Technol.*, **180**, *3*, 298 (2012); https://doi.org/10.13182/NT11-135.
16. J. TERRELL, "Distributions of Fission Neutron Numbers," *Phys. Rev.*, **108**, *3*, 783 (1957); https://doi.org/10.1103/PhysRev.108.783.
17. R. M. SAKIA, "The Box-Cox Transformation Technique: A Review," *J. R. Stat. Soc. Series D*, **41**, *2*, 169 (1992).
18. A. MARREL et al., "Calculations of Sobol Indices for the Gaussian Process Metamodel," *Reliab. Eng. Syst. Saf.*, **94**, *3*, 742 (2009); https://doi.org/10.1016/j.ress.2008.07.008.
19. N. METROPOLIS et al., "Equation of State Calculations by Fast Computing Machines," *J. Chem. Phys.*, **21**, *6*, 1087 (1953); https://doi.org/10.1063/1.1699114.
20. W. K. HASTINGS, "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika*, **57**, *1*, 97 (1970); https://doi.org/10.1093/biomet/57.1.97.
21. H. HAARIO, E. SAKSMAN, and J. TAMMINEN, "An Adaptive Metropolis Algorithm," *Bernoulli*, **7**, *2*, 223 (2001); https://doi.org/10.2307/3318737.
22. A. GELMAN, W. R. GILKS, and G. O. ROBERTS, "Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms," *Ann. Appl. Probab.*, **7**, *1*, 110 (1997); https://doi.org/10.1214/aoap/1034625254.
23. C. ANDRIEU and É. MOULINES, "On the Ergodicity Properties of Some Adaptive MCMC Algorithms," *Ann. Appl. Probab.*, **16**, *3*, 1462 (2006); https://doi.org/10.1214/105051606000000286.
24. H. HAARIO, E. SAKSMAN, and J. TAMMINEN, "Adaptive Proposal Distribution for Random Walk Metropolis Algorithm," *Comput. Stat.*, **14**, *3*, 375 (1999); https://doi.org/10.1007/s001800050022.
25. B. EFRON and R. J. TIBSHIRANI, *An Introduction to the Bootstrap*, CRC Press (1994).
26. D. HIGDON et al., "Combining Field Data and Computer Simulations for Calibration and Prediction," *SIAM J. Sci. Comput.*, **26**, *2*, 448 (2004); https://doi.org/10.1137/S1064827503426693.
27. J. B. BRIGGS, L. SCOTT, and A. NOURI, "The International Criticality Safety Benchmark Evaluation Project," *Nucl. Sci. Eng.*, **145**, *1*, 1 (2003); https://doi.org/10.13182/NSE03-14.
28. J. HUTCHINSON et al., "Validation of Statistical Uncertainties in Subcritical Benchmark Measurements: Part II—Measured Data," *Ann. Nucl. Energy*, **125**, 342 (2019); https://doi.org/10.1016/j.anucene.2018.10.021.
29. J. K. MATTINGLY, "Polyethylene-Reflected Plutonium Metal Sphere: Subcritical Neutron and Gamma Measurements," Sandia National Laboratories (2009).
30. H. JEFFREYS, "An Invariant Form for the Prior Probability in Estimation Problems," *Proc. R. Soc. London Ser. A.*, **186**, *1007*, 453 (1946).
31. P. LARTAUD, P. HUMBERT, and J. GARNIER, "Uncertainty Quantification in Neutron Noise Analysis Using Monte-Carlo Markov Chain Methods: An Application

to Nuclear Waste Drum Assay," *Proc. Int. Conf. on Physics of Reactors 2022 (PHYSOR 2022*), p. 2674 (2022).

32. K. KERSTING et al., "Most Likely Heteroscedastic Gaussian Process Regression," *Proc. 24th Int. Conf. on Machine Learning*, p. 393 (2007).

33. J. M. VERBEKE et al., "Correlated Production and Analog Transport of Fission Neutrons and Photons Using Fission Models FREYA, FIFRELIN and the Monte Carlo Code TRIPOLI-4®." *EPJ Web of Conf.*, Vol. 170, p. 01019, EDP Sciences (2018).