PCCP

PAPER

View Article Online
View Journal | View Issue

Cite this: *Phys. Chem. Chem. Phys.*, 2024, **26**, 3389

Log-Gaussian gamma processes for training Bayesian neural networks in Raman and CARS spectroscopies†

Teemu Härkönen, *[a] Erik M. Vartiainen, [a] Lasse Lensu, [a] Matthew T. Moores[ab] and Lassi Roininen[a]

We propose an approach utilizing gamma-distributed random variables, coupled with log-Gaussian modeling, to generate synthetic datasets suitable for training neural networks. This addresses the challenge of limited real observations in various applications. We apply this methodology to both Raman and coherent anti-Stokes Raman scattering (CARS) spectra, using experimental spectra to estimate gamma process parameters. Parameter estimation is performed using Markov chain Monte Carlo methods, yielding a full Bayesian posterior distribution for the model which can be sampled for synthetic data generation. Additionally, we model the additive and multiplicative background functions for Raman and CARS with Gaussian processes. We train two Bayesian neural networks to estimate parameters of the gamma process which can then be used to estimate the underlying Raman spectrum and simultaneously provide uncertainty through the estimation of parameters of a probability distribution. We apply the trained Bayesian neural networks to experimental Raman spectra of phthalocyanine blue, aniline black, naphthol red, and red 264 pigments and also to experimental CARS spectra of adenosine phosphate, fructose, glucose, and sucrose. The results agree with deterministic point estimates for the underlying Raman and CARS spectral signatures.

Received 12th October 2023,
Accepted 3rd January 2024

DOI: 10.1039/d3cp04960d

rsc.li/pccp

1 Introduction

Raman and coherent anti-Stokes Raman scattering (CARS) spectroscopies are vital tools used in chemistry, physics, and biomedical research.[1–3] The insights they offer into molecular vibrations, structural dynamics, and chemical compositions are invaluable. However, working with their data presents challenges. Measurement artifacts including noise, and, especially, background signals in Raman and CARS spectra often obscure crucial molecular information. Traditional methods for data correction are typically manual and may fall short in capturing the full complexity of the data. For instance, standard approaches used for removing the background signals include asymmetric least squares polynomial fitting, wavelet-based methods, optimization with Tikhonov regularization, and Kramers–Kronig relations.[4–12] While appealing, these methods suffer from practical drawbacks such as the need for manual tuning of the model or regularization parameters. The need for automated, robust, and statistically sound solutions to enhance our spectroscopic analyses is evident.

Deep neural networks offer a compelling solution for automatic spectral correction across various applications, from weather predictions[13–15] to medical imaging[16–18] and many others.[19–23] In the realm of Raman spectroscopy, deep neural networks have been used in chemical species identification and background removal.[24–28] Similarly, they have been applied to extract the underlying Raman spectra from CARS measurement.[29–35] Despite their efficacy, non-Bayesian neural networks lack a critical feature: the ability to quantify uncertainty in Raman spectrum estimation. Bayesian inference, on the other hand, provides an avenue to solve this problem.

Bayesian inference treats the parameters of a given model as random variables. These models consist of a likelihood function that is combined with prior distributions for the parameters to produce posterior estimates. The likelihood function is analogous to a utility function in an optimization context. It quantifies how well the model fits the observed data. The aforementioned prior distributions for the model parameters represent the information known beforehand, including any constraints dictated by the physical nature of the parameters, such as non-negativity. In spectroscopic analysis, the model

[a] *Department of Computational Engineering, School of Engineering Sciences, LUT University, Yliopistonkatu 34, FI-53850, Lappeenranta, Finland. E-mail: teemu.harkonen@lut.fi*
[b] *National Institute for Applied Statistics Research Australia, University of Wollongong, Wollongong, NSW 2522, Australia*

† Electronic supplementary information (ESI) available. See DOI: https://doi.org/10.1039/d3cp04960d

parameters can be, for example, amplitudes, locations, and widths of Gaussian, Lorentzian, or Voigt line shape functions. The combination of the likelihood and the priors results in a posterior distribution over the model parameters. The posterior is a probabilistic representation of the uncertainty in the parameter estimates. Bayesian approaches have been considered for estimating spectrum parameters, where the authors used sequential Monte Carlo algorithms to numerically sample from the posterior distribution.[36,37] While the uncertainty quantification provided by Bayesian modeling and Markov chain Monte Carlo (MCMC) methods is compelling, the approach is known to be computationally expensive, see for example.[38] This becomes a major issue particularly with hyperspectral data sets. A hyperspectral data set, or an image, consists of pixels where each pixel contains a spectrum. This can quickly result in millions of individual spectra, experimental or synthetic, which are to be analyzed.

Bayesian neural networks are a synthesis of the aforementioned two ideas. Bayesian neural networks model the weights and biases of standard neural networks as random variables, which can be assigned prior distributions. When combined with a likelihood according to Bayes' theorem, the resulting utility function corresponds to the posterior for the neural network parameters. Advantages of this Bayesian neural network approach in comparison to non-Bayesian neural networks include robustness in terms of overfitting, providing uncertainty estimates instead of only point estimation, sequential learning, and better generalization.[39] In particular, uncertainty quantification has seen widespread research covering many application areas and topics, for example.[40]

One of the challenges of Bayesian neural networks is that they typically contain an enormous number of parameters. For instance, our network comprises over 11 million parameters, far beyond what is commonly considered high-dimensional for MCMC.[41,42] Some neural networks, such as large language models (LLMs), can have billions of parameters.[43] Thus, it can be challenging to establish convergence of such a large number of parameters in a statistically rigorous manner. To combat this, partially-Bayesian neural networks have been used as a practical tool to provide uncertainty estimation with neural networks. In addition to empirical validation through practice, studies have provided compelling analytical and numerical evidence that partially-Bayesian neural networks are indeed capable of providing posterior estimates on par or even superior performance to fully-Bayesian neural networks.[44] The above points lead us to construct our neural network for this study as a partially-Bayesian neural network.

Neural networks typically require large volumes of training data. This has been noted to be a problem also in spectroscopic applications as it is difficult to acquire large sets of independent data sets.[28] Therefore, many studies mentioned above use synthetic data to train the neural networks. The synthetic data is usually generated using random linear combinations of Lorentzian line shapes, where the amplitudes, locations, and widths are sampled from predefined probability distributions, see for example.[25,29,45] The background data is generated

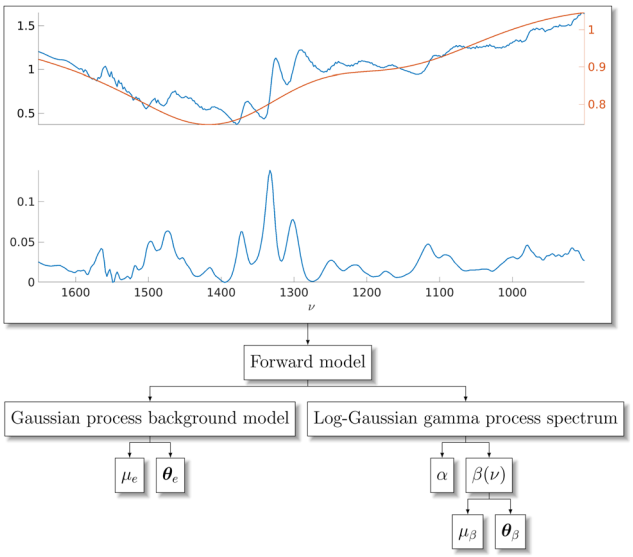

Fig. 1 Structure of our generative spectrum model using GPs and log-Gaussian gamma processes. On top, an experimental CARS spectrum of adenosine phosphate in blue and an example multiplicative background in red. We model the backgrounds as a GP. At the bottom, an example underlying Raman spectral signature in blue. We assume the Raman peaks to be distributed according to our proposed log-Gaussian gamma process model. The stochastic processes are parameterized according to μ_e, $\boldsymbol{\theta}_e$, α, and $\beta(\nu)$. We further model $\beta(\nu)$ using GPs which are parameterized according to μ_β and $\boldsymbol{\theta}_\beta$. We construct statistical samples with MCMC for the model parameters which allow us to generate synthetic spectra for training our Bayesian neural network.

similarly. The backgrounds are modeled explicitly using a parametric functional form, such as a polynomial or a sigmoidal function, and the parameters of the model are again sampled from a predefined probability distribution.[25,32,46] An extension to this is to use experimental Raman spectra on top of the randomly generated spectra.[34]

Stochastic processes can be used to draw samples of random functions. A typical example of a stochastic process is the widely-used Gaussian process (GP). Properties of the drawn samples such as differentiability are governed through kernel functions, which are used to model dependencies between data points. For readers unfamiliar with GPs, we recommend the book by Rasmussen and Williams.[47] Instead of using explicit, parametric functions to model the spectroscopic features, we propose using stochastic processes as a more flexible tool for the purpose. In this study, we use GPs as a generative model for the additive and multiplicative backgrounds of Raman and CARS spectra, see Fig. 1.

For the purpose of generating synthetic Raman spectral signatures, we propose a specific type of doubly-stochastic Lévy process which we call a log-Gaussian gamma process. Our construction of the log-Gaussian gamma process is inspired by log-Gaussian Cox process which the authors have previously used as a model for spectra.[48] While it makes sense to model spectra as a Cox process where the relaxation from higher energy levels happens at a constant rate and results in counts of photons, the data is often available in scaled floating-point

numbers which prevents direct application of the log-Gaussian Cox process model. Gamma-distributed variables have direct connections to Poisson-distributed variables, which constitute the Cox process, making the extension to a log-Gaussian gamma process intuitive as a model for Raman spectroscopy. The log-Gaussian gamma process can be used to generate arbitrary amounts of synthetic spectra once parameters of the stochastic process have been estimated. We perform the estimation using MCMC methods which allow us to construct a Bayesian posterior distribution for the model parameters, thereby including uncertainty of the parameter estimates in our data generation. This also applies to our GP-based background model. We present a high-level diagram of our stochastic process method for data generation in Fig. 1. Fig. 2 shows an example of the aim of this paper, a Raman spectral signature extracted from a CARS spectrum using a Bayesian neural network. We provide a pseudo-code description of our approach in Algorithm 3.

The key contributions of this paper are the following. We propose using log-Gaussian gamma processes for modeling Raman spectral signatures and GPs to model additive or multiplicative background signals. The aforementioned doubly-stochastic processes are sampled randomly, enabling us to generate an arbitrary number of synthetic spectra that are statistically similar to experimental spectra. Finally, we present a partially-Bayesian neural network for analyzing Raman and CARS spectral measurements, which we train using the sampled synthetic spectra. Once trained, we use these neural networks to estimate the spectral signatures for experimental Raman spectroscopy measurements of phthalocyanine blue, naphthol red, aniline black, and red 264 pigments and for experimental CARS spectra of adenosine phosphate, fructose, glucose, and sucrose in addition to synthetic test spectra.

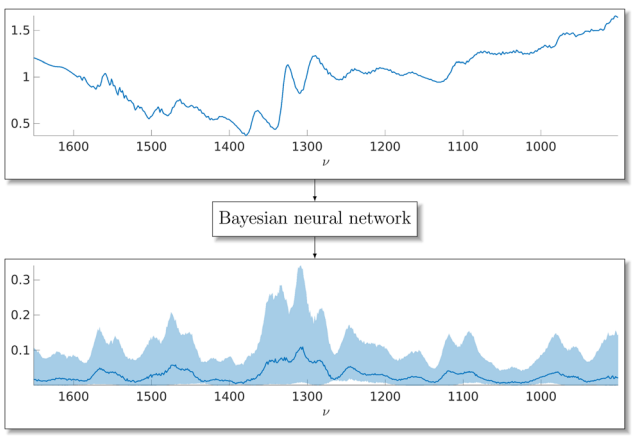

Fig. 2 On top, an experimental CARS spectrum of adenosine phosphate in blue. With a trained Bayesian neural network, we can extract the underlying Raman spectral signature from the data along with an uncertainty estimate for the spectrum. At the bottom, the corresponding Bayesian neural network median Raman spectrum estimate and 90% confidence interval of the estimate for the adenosine phosphate data.

Algorithm 1 Log-Gaussian gamma process data generation for training Bayesian neural networks

Step 1: Fit a log-Gaussian gamma process to Raman spectrum data.
Step 2: Fit a GP to background data.
Step 3: Draw a large number of realizations from the fitted log-Gaussian gamma process.
Step 4: Draw a large number of realizations from the fitted GP.
Step 5: Use a forward model to combine the realizations to form a data set of synthetic spectra.
Step 6: Use a forward model to combine the realizations to form a data set of synthetic spectra.

The rest of the paper is structured as follows. We detail the steps used to generate the synthetic training data in three stages in the following three sections. We first present the log-Gaussian gamma process as a model for Raman spectral signatures and explain how to draw realizations of this doubly-stochastic process. This is followed by a description of our GP-based additive and multiplicative background models. We finalize the explanation of our synthetic data generation method with definitions of the forward models used to simulate synthetic training data for Raman and CARS measurements with additive and multiplicative backgrounds, respectively. Next, we present our partially-Bayesian neural network architecture, which we train against the synthetic data sets that we have generated. We document computational details and prior distributions in the next section, followed by a presentation of our results for both artificial and real experimental data. Finally, we conclude with a discussion of the significance and other potential applications for our method.

2 Log-Gaussian gamma process spectrum model

We model a Raman spectral signature as a collection of conditionally-independent, gamma-distributed random variables

$$r_k := r(\nu_k) \sim \text{Gamma}(\alpha, \beta(\nu_k)), \quad (1)$$

where r_k denotes a Raman measurement at wavenumber location ν_k with α and $\beta(\nu_k)$ being the shape and scale parameters of the gamma distribution, respectively. The scale $\beta(\nu_k)$ is thought to model the spectral line shapes and other artefacts, while the shape α models the noise level present in the spectrum. The above construction is motivated by log-Gaussian Cox processes[49] but without the restriction of modeling of only integer-valued data and with an additional parameter in the stochastic process allowing for more flexible modeling of uncertainty. Poisson-distributed random variables, which constitute the Cox process, have a single parameter to control both the mean and variance of the distribution. Very often in real data, this assumption is found to be too restrictive, leading to a

model that is either under- or over-dispersed.[50] In contrast, the gamma distribution has two parameters which together allow for a range of different variances for a given mean.

We extend eqn (1) by modeling the log-scale as a GP, resulting in a hierarchical model

$$\log \beta(\boldsymbol{\nu}) \sim \mathrm{GP}(\mu_\beta, \Sigma_\beta(\boldsymbol{\nu},\boldsymbol{\nu},\boldsymbol{\theta}_\beta)), \tag{2}$$

where $\boldsymbol{\nu}$: = $(\nu_1,\ldots,\nu_K)$ is a vector of the wavenumber locations with $\mu_\beta \in \mathbb{R}$ and $\Sigma_\beta(\boldsymbol{\nu},\boldsymbol{\nu},\boldsymbol{\theta}_\beta) \in \mathbb{R}^{K\times K}$ being a constant mean and a covariance matrix parameterized according to hyperparameters $\boldsymbol{\theta}_\beta$. This doubly-stochastic model introduces dependence between values r_i and r_j at different wavenumbers ν_i and ν_j. For the covariance function of the log-scale GP, we use the squared exponential kernel

$$\left[\Sigma_\beta(\boldsymbol{\nu},\boldsymbol{\nu},\boldsymbol{\theta}_\beta)\right]_{ij} = \sigma_{\beta,f}^2 \exp\left(-\frac{1}{2}\frac{(\nu_i-\nu_j)^2}{l_\beta^2}\right) + \sigma_\beta^2 \delta(\nu_i-\nu_j), \tag{3}$$

where $\left[\Sigma_e(\boldsymbol{\nu},\boldsymbol{\nu},\boldsymbol{\theta}_\beta)\right]_{ij}$ denotes the ijth element of the covariance matrix $\Sigma_\beta(\boldsymbol{\nu},\boldsymbol{\nu},\boldsymbol{\theta}_\beta)$, ${\sigma_\beta}_f^2$ is the signal variance, l_β is the length scale, ${\sigma_\beta}^2$ denotes the noise variance, and $\delta(\nu_i-\nu_j)$ is the Dirac delta function with $\boldsymbol{\theta}_\beta = ({\sigma_\beta}_f^2, l_\beta, {\sigma_\beta}^2)$. The GP construction yields an analytical form for the log-scale $\log\beta(\boldsymbol{\nu})$ which we will detail below as we construct the posterior distribution according to Bayes' theorem. This log-GP parameterization is identical to the log-intensity model for Poisson variables that features in log-Gaussian Cox processes. For more details on the log-Gaussian Cox process, see[49] and for example.[51]

The posterior distribution involves the likelihood function $\mathscr{L}(\boldsymbol{r}|\alpha,\beta(\boldsymbol{\nu}))$, a log-GP prior for the scale $\pi_0(\beta(\boldsymbol{\nu})|\mu_\beta,\boldsymbol{\theta}_\beta)$, and a joint prior distribution $\pi_0(\alpha,\mu_\beta,\boldsymbol{\theta}_\beta)$ for rest of the model parameters. Given a measured Raman spectrum $\boldsymbol{r}$: = $(r(\nu_1),\ldots,r(\nu_K))^{\mathrm{T}}$, we can formulate the likelihood as a product of conditionally-independent, gamma-distributed random variables

$$\mathscr{L}(\boldsymbol{r}|\alpha,\beta(\boldsymbol{\nu})) \propto \prod_{k=1}^{K} \frac{r_k^{\alpha-1}\exp(-r_k/\beta_k)}{\Gamma(\alpha)\beta_k^\alpha}, \tag{4}$$

where β_k: =$\beta(\nu_k)$, and $\Gamma(\alpha)$ is the gamma function. The hierarchical prior for $\beta(\boldsymbol{\nu})$ can be evaluated as

$$\pi_0(\beta(\boldsymbol{\nu})|\mu_\beta,\boldsymbol{\theta}_\beta) = \frac{1}{\sqrt{(2\pi)^K}}\left|\Sigma_\beta(\boldsymbol{\nu},\boldsymbol{\nu};\mathbf{q}_\beta)\right|^{-1/2} \times \exp\left(-\frac{1}{2}(\beta(\boldsymbol{\nu})-\mu_\beta)^{\mathrm{T}}\Sigma_\beta(\boldsymbol{\nu},\boldsymbol{\nu},\boldsymbol{\theta}_\beta)^{-1}(\beta(\boldsymbol{\nu})-\mu_\beta)\right), \tag{5}$$

where $|\Sigma_\beta(\boldsymbol{\nu},\boldsymbol{\nu};\boldsymbol{\theta}_\beta)|$ denotes the determinant of the covariance matrix. With the above and a joint prior $\pi_0(\alpha,\mu_\beta,\boldsymbol{\theta}_\beta)$, we can construct the posterior distribution for the model parameters conditioned on the measured spectrum data $\boldsymbol{r}$ as

$$\pi(\alpha,\beta(\boldsymbol{\nu}),\mu_\beta,\boldsymbol{\theta}_\beta|\boldsymbol{r}) \propto \mathcal{L}(\boldsymbol{r}|\alpha,\beta(\boldsymbol{\nu})\pi_0(\beta(\boldsymbol{\nu}))\mu_\beta,\boldsymbol{\theta}_\beta) \times \pi_0(\alpha,\mu_\beta,\boldsymbol{\theta}_\beta). \tag{6}$$

In the posterior in eqn (6), the dimension of $\beta(\boldsymbol{\nu})$ is K.

The scale is a vector of the same dimension as the data, $\beta(\boldsymbol{\nu}) \in \mathbb{R}_+^{K\times 1}$. MCMC methdos are known to struggle estimating high-dimensional parameters. At a minimum, the high-dimensional parameters incur a computational cost for inference with MCMC. To amend these issues and to simplify the inference, we perform dimension reduction for the scale $\beta(\boldsymbol{\nu})$. To achieve this, we observe that our data $\boldsymbol{r}$ should be a reasonable estimate for the expectation of the gamma process in eqn (1), $\boldsymbol{r} \approx \mathbb{E}[\mathrm{Gamma}(\alpha,\beta(\boldsymbol{\nu}))] = \alpha\beta(\boldsymbol{\nu})$. This implies that the shape of the data $\boldsymbol{r}$ is close to the shape of the scale function, $\beta(\boldsymbol{\nu})$. Thus, we approximate the scale $\beta(\boldsymbol{\nu})$ as a convolution between a Gaussian kernel and the data

$$\beta(\boldsymbol{\nu}) \approx c_\beta G(\boldsymbol{\nu};\sigma_G) * \boldsymbol{r}, \tag{7}$$

where $*$ denotes convolution, c_β is a scaling constant, and $G(\boldsymbol{\nu};\sigma_G)$ is Gaussian smoothing kernel with width σ_G. By this, we reduce the inference of the scale $\beta(\boldsymbol{\nu}) \in \mathbb{R}_+^{K\times 1}$ to inference of two parameters, c_β and σ_G. With this smoothing approximation, we formulate an approximate posterior for eqn (6) as

$$\pi(\alpha,c_\beta,\sigma_G,\mu_\beta,\boldsymbol{\theta}_\beta|\boldsymbol{r}) = \widetilde{\mathscr{L}}(\boldsymbol{r}|\alpha,c_\beta,\sigma_G)\pi_0(\beta(\boldsymbol{\nu})|\mu_\beta,\boldsymbol{\theta}_\beta) \times \pi_0(\alpha,c_\beta,\sigma_G,\mu_\beta,\boldsymbol{\theta}_\beta), \tag{8}$$

where $\widetilde{\mathscr{L}}(\boldsymbol{r}|\alpha,c_\beta,\sigma_G) = \widetilde{\mathscr{L}}(\boldsymbol{r}|\alpha,c_\beta,G(\boldsymbol{\nu};\sigma_G)*\boldsymbol{r})$ and $\pi_0(\alpha,c_\beta,\sigma_G,\mu_\beta,\boldsymbol{\theta}_\beta)$ is the prior distribution augmented with $(c_\beta,\sigma_G)^{\mathrm{T}}$. We detail the prior distribution $\pi_0(\alpha,c_\beta,\sigma_G,\mu_\beta,\boldsymbol{\theta}_\beta)$ in the section on computational details and prior distributions. We perform inference of the posterior in eqn (8) by sampling all the model parameters simultaneously using the DRAM algorithm.[52]

Given samples from the posterior distribution $\pi(\alpha,c_\beta,\sigma_G,\mu_\beta,\boldsymbol{\theta}_\beta\,|\,\boldsymbol{r})$ obtained with MCMC, we can sample realizations for the synthetic spectra to generate an arbitrary amount of synthetic data in the following way. First, we sample the GP parameters $(\tilde{\mu}_\beta\tilde{\boldsymbol{\theta}}_\beta)^{\mathrm{T}}$ from the MCMC chain. Next, we use $(\tilde{\mu}_\beta\tilde{\boldsymbol{\theta}}_\beta)^{\mathrm{T}}$ sample a GP realization $\tilde{\beta}(\boldsymbol{\nu}^\star|\tilde{\mu}_\beta\tilde{\boldsymbol{\theta}}_\beta)$ at prediction locations $\boldsymbol{\nu}^*$: = $(\nu_1^*,\ldots,\nu_{\tilde{K}}^*)^{\mathrm{T}}$ modeling the scale $\beta(\boldsymbol{\nu}^\star)$ with

$$\tilde{\beta}(\boldsymbol{\nu}^\star|\tilde{\mu}_\beta\tilde{\boldsymbol{\theta}}_\beta) = \exp(\tilde{\mu}_\beta + L(\boldsymbol{\nu}^\star|\tilde{\boldsymbol{\theta}}_\beta)\boldsymbol{u}), \tag{9}$$

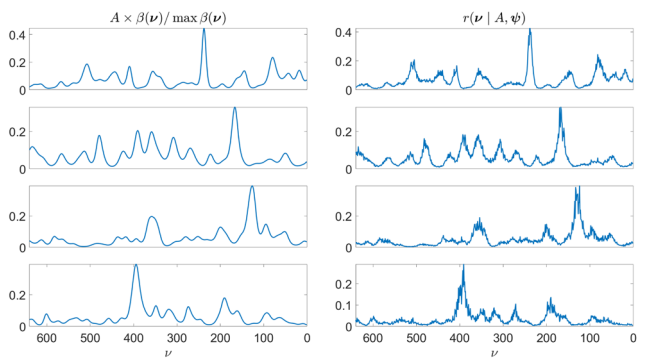

Fig. 3 Example realizations drawn from the log-Gaussian gamma process model defined in eqn (10). On the left, realizations for the scale process $\beta(\boldsymbol{\nu})$, drawn from a log-Gaussian process. On the right, corresponding realizations from the gamma process. All realizations are normalized and multiplied by a sampled amplitude.

where $L(\boldsymbol{\nu}^\star \mid \tilde{\boldsymbol{\theta}}_\beta)$ is the lower triangular Cholesky decomposition matrix of $\Sigma_\beta(\boldsymbol{\nu}^\star,\boldsymbol{\nu}^\star,\tilde{\boldsymbol{\theta}}_\beta)$ and $\boldsymbol{u}$: = $(u_1,\ldots,u_{\tilde{K}})^{\mathrm{T}}$ is Gaussian white noise such that $u_{\tilde{k}} \sim \mathcal{N}(0,1)$. Finally, by sampling $\tilde{\alpha}$, we can draw a spectrum realization $\tilde{r}(\boldsymbol{\nu})$ from the gamma process, Gamma$(\tilde{\alpha},\tilde{\beta}(\boldsymbol{\nu}))$.

We normalize the realizations $\tilde{r}(\boldsymbol{\nu})$ such that $\max\{\tilde{r}(\boldsymbol{\nu})\} = 1$ and introduce an additional parameter to control amplitudes of the realizations. With an amplitude parameter A, we sample a normalized shape $\tilde{r}_{\mathrm{N}}(\boldsymbol{\nu} \mid \boldsymbol{\psi})$ of the spectrum and multiply this by a sampled amplitude $\tilde{A}$. This procedure results in the following statistical model

$$\begin{aligned} r(\boldsymbol{\nu}|A,\boldsymbol{\psi}) &\sim A\frac{r_N(\boldsymbol{\nu}|\boldsymbol{\psi})}{\max r_N(\boldsymbol{\nu}|\boldsymbol{\psi})},\\ r_N(\boldsymbol{\nu}|\boldsymbol{\psi}) &\sim \mathrm{Gamma}(\alpha,\beta(\boldsymbol{\nu})),\\ A &\sim \pi_0(A),\\ \boldsymbol{\psi} &\sim \pi_0(\boldsymbol{\psi}), \end{aligned} \tag{10}$$

where $\boldsymbol{\psi}$: = $(\alpha,c_\beta,\sigma_G,\mu_\beta,\boldsymbol{\theta}_\beta)^{\mathrm{T}}$ is a shorthand for the gamma process parameters and $\pi_0(A)$ is a prior distribution for the amplitude A. Example realizations from the above statistical model are shown in Fig. 3. In the following section, we detail how we model additive and multiplicative backgrounds for Raman and CARS spectra using GPs.

3 Additive and multiplicative background models

We propose GPs as a flexible way to randomly draw additive and multiplicative background functions for Raman and CARS spectrum modeling. This is in contrast to more standard polynomial models such as the ones used in.[32]

As noted above, we model additive or multiplicative spectral backgrounds as a GP

$$e(\boldsymbol{\nu}) \sim \mathrm{GP}(\mu_e,\Sigma_e(\boldsymbol{\nu},\boldsymbol{\nu},\boldsymbol{\theta}_e)), \tag{11}$$

with $\mu_e \in \mathbb{R}$ and $\Sigma_e(\boldsymbol{\nu},\boldsymbol{\nu},\boldsymbol{\theta}_e) \in \mathbb{R}^{K\times K}$ being a constant mean and the covariance matrix of the GP parameterized according to hyperparameters $\boldsymbol{\theta}_e$. For the background GP covariance function, we use again the squared exponential kernel

$$[\Sigma_e(\boldsymbol{\nu},\boldsymbol{\nu},\boldsymbol{\theta}_e)]_{i,j} = \sigma_{e,f}^2 \exp\left(-\frac{1}{2}\frac{(\nu_i-\nu_j)^2}{l_e^2}\right) + \sigma_e^2\delta(\nu_i-\nu_j) \tag{12}$$

where $[\Sigma_e(\boldsymbol{\nu},\boldsymbol{\nu},\boldsymbol{\theta}_e)]_{i,j}$ denotes the ijth element of the covariance matrix, ${\sigma_{e,f}}^2$ is the signal variance, l_e is the length scale, and ${\sigma_e}^2$ denotes the noise variance with $\boldsymbol{\theta}_e$: = $(\sigma_{e,f},l_e,\sigma_e)^{\mathrm{T}}$.

Given a measurement of the background process, $\boldsymbol{e}$: = $(e(\nu_1),\ldots,e(\nu_K))^{\mathrm{T}}$, we can formulate a posterior distribution for the background GP parameters $(\mu_e,\boldsymbol{\theta}_e)^{\mathrm{T}}$ as

$$\pi(\mu_e,\boldsymbol{\theta}_e|\boldsymbol{e}) \propto \mathcal{L}(\boldsymbol{e}|\mu_e,\boldsymbol{\theta}_e)\pi_0(\mu_e,\boldsymbol{\theta}_e), \tag{13}$$

where $\mathcal{L}(\boldsymbol{e}|\mu_e,\boldsymbol{\theta}_e)$ is the GP likelihood and $\pi_0(\mu_e,\boldsymbol{\theta}_e)$ denotes the prior distribution for the GP parameters. The log-likelihood is given as

$$\begin{aligned} \log\mathcal{L}(\boldsymbol{e}|\mu_e,\boldsymbol{\theta}_e) &= -\frac{1}{2}(\boldsymbol{e}-\mu_e)^{\mathrm{T}}\Sigma_e(\boldsymbol{\nu},\boldsymbol{\nu},\theta_e)^{-1}(\boldsymbol{e}-\mu_e)\\ &\quad -\frac{1}{2}\log|\Sigma_e(\boldsymbol{\nu},\boldsymbol{\nu},\theta_e)| - \frac{K}{2}\log 2\pi, \end{aligned} \tag{14}$$

where $|\Sigma_e(\boldsymbol{\nu},\boldsymbol{\nu},\boldsymbol{\theta}_e)|$ is the determinant of the covariance matrix. Again, we perform the posterior estimation for eqn (13) by sampling all the model parameters simultaneously using DRAM.

Given a posterior $\pi(\mu_e,\boldsymbol{\theta}_e \mid \boldsymbol{e})$, we construct realizations for the spectrum by drawing realizations from the GP predictive distribution. We sample starting and ending points for the background function from priors $\pi_0(e_{\mathrm{start}})$ and $\pi_0(e_{\mathrm{stop}})$, $\pi_0(e_{\mathrm{start}},e_{\mathrm{stop}}) = \pi_0(e_{\mathrm{start}})\pi_0(e_{\mathrm{stop}})$. Next, we compute the predictive mean

$$e^\star(\boldsymbol{\nu} \mid \tilde{\mu}_e,\tilde{\boldsymbol{\theta}}_e,\tilde{\varepsilon}_{\mathrm{ss}}) = \Sigma_e(\boldsymbol{\nu},\boldsymbol{\nu}_{\mathrm{ss}};\tilde{\boldsymbol{\theta}}_e)\Sigma_e(\boldsymbol{\nu}_{\mathrm{ss}},\boldsymbol{\nu}_{\mathrm{ss}};\tilde{\boldsymbol{\theta}}_e)^{-1} \times (\tilde{\boldsymbol{e}}_{\mathrm{ss}} - \tilde{\mu}_e) + \tilde{\mu}_e, \tag{15}$$

and the predictive covariance

$$\begin{aligned} \Sigma_e^*\left(\boldsymbol{\nu},\boldsymbol{\nu};\tilde{\boldsymbol{\theta}}_e\right) &= \Sigma_e\left(\boldsymbol{\nu},\boldsymbol{\nu};\tilde{\boldsymbol{\theta}}_e\right) - \Sigma_e\left(\boldsymbol{\nu},\boldsymbol{\nu}_{\mathrm{ss}};\tilde{\boldsymbol{\theta}}_e\right)\Sigma_e\left(\boldsymbol{\nu}_{\mathrm{ss}},\boldsymbol{\nu}_{\mathrm{ss}};\tilde{\boldsymbol{\theta}}_e\right)^{-1}\\ &\quad \times\Sigma_e\left(\boldsymbol{\nu},\boldsymbol{\nu}_{\mathrm{ss}};\tilde{\boldsymbol{\theta}}_e\right)^{\mathrm{T}}, \end{aligned} \tag{16}$$

where $(\tilde{\mu}_e,\tilde{\boldsymbol{\theta}}_e)$ are samples from the posterior distribution $\pi(\mu_e,\boldsymbol{\theta}_e \mid \boldsymbol{e})$ obtained *via* MCMC, and $\boldsymbol{\nu}_{\mathrm{ss}}$= $(\nu_{\mathrm{start}},\nu_{\mathrm{stop}})$ are the wavenumber locations corresponding to the sampled starting and ending points $\tilde{\boldsymbol{e}}_{\mathrm{ss}} = (\tilde{e}_{\mathrm{start}},\tilde{e}_{\mathrm{stop}})^{\mathrm{T}}$. Elements of the covariance matrix $\Sigma_e(\boldsymbol{\nu},\boldsymbol{\nu};\tilde{\boldsymbol{\theta}}_e)$ are given as defined in eqn (12) and elements of the covariance matrices $\Sigma_e(\boldsymbol{\nu},\boldsymbol{\nu}_{\mathrm{ss}};\tilde{\boldsymbol{\theta}}_e)$ and $\Sigma_e(\boldsymbol{\nu}_{\mathrm{ss}},\boldsymbol{\nu}_{\mathrm{ss}};\tilde{\boldsymbol{\theta}}_e)$ are given by otherwise the same covariance function but without the diagonal elements produced by the Dirac delta function.

With the above mathematical machinations, we can sample realizations for the background function by

$$\tilde{e}(\boldsymbol{\nu} \mid \tilde{\mu}_e,\tilde{\boldsymbol{\theta}}_e,\tilde{\boldsymbol{e}}_{\mathrm{ss}}) \sim e^\star(\boldsymbol{\nu} \mid \tilde{\mu}_e,\tilde{\boldsymbol{\theta}}_e,\tilde{\boldsymbol{e}}_{\mathrm{ss}}) + L(\boldsymbol{\nu} \mid \tilde{\mu}_e,\tilde{\boldsymbol{\theta}}_e)\boldsymbol{w}, \tag{17}$$

where $L(\boldsymbol{\nu} \mid \tilde{\mu}_e,\tilde{\boldsymbol{\theta}}_e)$ is the lower triangular Cholesky decomposition matrix of $\Sigma^\star(\boldsymbol{\nu},\boldsymbol{\nu};\tilde{\boldsymbol{\theta}}_e)$ and $\boldsymbol{w} \in \mathbb{R}^{K\times 1}$ is a Gaussian white noise vector. This is compiled into the following statistical

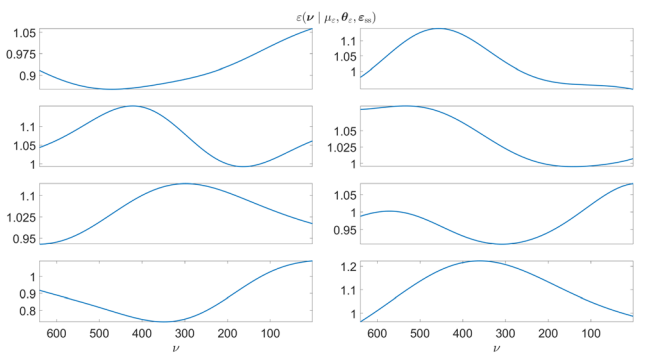

Fig. 4 Example realizations drawn from the background model defined in eqn (18) for a multiplicative background. The starting and end points are sampled from a prior distribution and the GP predictive mean and covariance are used to sample the background shape.

model for the background function sampling:

$$\tilde{e}(\boldsymbol{v}|\mu_e, \boldsymbol{\theta}_e, \boldsymbol{e}_{\mathrm{ss}}) \sim \mathcal{N}(\varepsilon^*, \Sigma^*),$$
$$(\mu_e, \boldsymbol{\theta}_e, \boldsymbol{e}_{\mathrm{ss}}) \sim \pi(\mu_e, \boldsymbol{\theta}_e|\boldsymbol{e})\pi_0(\boldsymbol{e}_{\mathrm{ss}}), \quad (18)$$

where $\pi_0(\boldsymbol{e}_{\mathrm{ss}}) = \pi_0(e_{\mathrm{start}}, e_{\mathrm{stop}})$. Example realizations for a multiplicative background relevant for CARS are shown in Fig. 4.

4 Raman and CARS spectrum models

In the preceding two sections we formulated mathematical procedures to sample synthetic spectrum and background realizations which are statistically similar to measurement data. Below, we combine these two approaches for generating arbitrary amounts of statistically realistic spectral data which are ultimately used for training our Bayesian neural networks. We present two forward models which are used to generate data for Raman measurements with an additive background and CARS measurements with a multiplicative background.

Raman spectra $y(\nu)$ with an additive background $B(\nu)$ are constructed using

$$y(\nu) \sim r(\nu \,|\, A, \psi) + B(\nu), \quad (19)$$

where $r(\nu|A,\psi)$ is distributed according to the model defined in eqn (10). The background $B(\nu)$ is sampled with eqn (18).

CARS spectra $z(\nu)$ are generated similarly to the additive Raman realizations. The CARS model consists of a multiplicative background function $\varepsilon_{\mathrm{m}}(\nu \,|\, \mu_e, \boldsymbol{\theta}_e)$ distorting a CARS spectrum $S(\nu|B_{\mathrm{NR}}, \psi)$ given as

$$z(\nu) \sim \varepsilon_{\mathrm{m}}(\nu|\mu_e, \boldsymbol{\theta}_e) S(\nu|B_{\mathrm{NR}}, \psi), \quad (20)$$

where the CARS spectrum $S(\nu|B_{\mathrm{NR}}, \psi)$ can be given as

$$S(\nu|B_{\mathrm{NR}}, \psi) \sim |B_{\mathrm{NR}} + (\mathrm{i}r(\nu|A, \psi) - \mathcal{H}\{r(\nu|A, \psi)\})|^2, \quad (21)$$

and $B_{\mathrm{NR}} \sim \pi_0(B_{\mathrm{NR}})$ is a non-resonant background inherent to the CARS phenomenon distributed according to a prior distribution $\pi_0(B_{\mathrm{NR}})$ and $\mathcal{H}$ denotes the Hilbert transform. The model for the CARS spectrum has been previously used for example in.[9,37] We show example realizations for the Raman model in Fig. 5 and the CARS model in Fig. 6. We use the two models defined in eqn (19) and (20) to generate two synthetic data sets which are used to train two separate Bayesian neural networks. In the following section, we discuss the Bayesian neural network architecture.

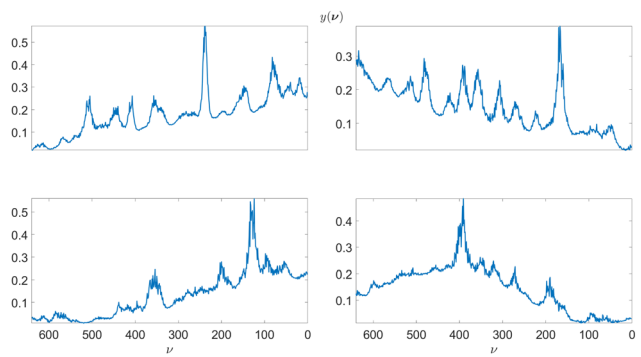

Fig. 5 Example realizations for the Raman spectrum model defined in eqn (19). The realizations correspond to the log-Gaussian gamma process realizations in Fig. 3.

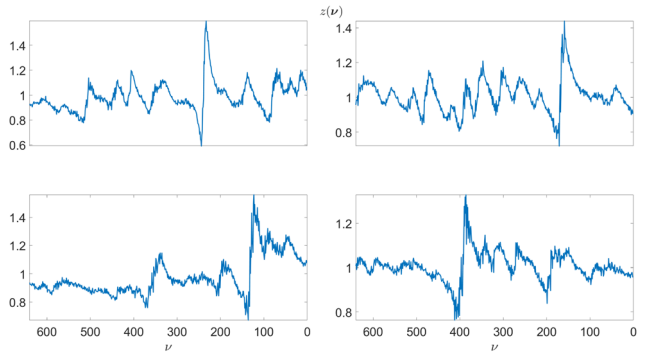

Fig. 6 Example realizations for the CARS spectrum model defined in eqn (20). The realizations correspond to the log-Gaussian gamma process realizations in Fig. 3.

5 Bayesian neural network architecture

Our neural network architecture used in the experiments is based on the SpecNet architecture.[29] The SpecNet architecture is composed of convolutional layers encoding the input, the measurement spectrum. The encoded information is then decoded using fully-connected hidden layers, resulting in estimates for the underlying true Raman spectrum. We present our changes to the SpecNet architecture below.

To achieve a partially Bayesian neural network,[44] we use a Bayesian layer for the first convolutional layer. Additionally, we augment the architecture with a probabilistic output layer. This transforms the neural network estimate into estimates of a stochastic process instead of directly estimating the Raman spectrum. We use a gamma distribution as our output layer, following our formulation of Raman spectra as a log-Gaussian gamma processes. We also found that L_1 or L_2 regularization was not necessary for the deterministic parts of the network and therefore only employ Dropout[53] regularization with the last dense layer of the network. This in agreement with the documented robustness of Bayesian neural networks with respect to overfitting.[39] The above results in the following partial posterior probability distribution, or cost function, used for training the neural network

$$\pi(\Psi_{\mathrm{D}}, \Psi_{\mathrm{S}}|R) \propto \mathcal{L}(R|\Psi_{\mathrm{D}}, \Psi_{\mathrm{S}})\pi_0(\Psi_{\mathrm{S}}), \quad (22)$$

where $R \in \mathbb{R}^{I \times J}$ is a data matrix of I synthetic spectra of length J generated using either the Raman or CARS forward models in eqn (19) and (20) and $\mathcal{L}(R|\Psi_{\mathrm{D}}, \Psi_{\mathrm{S}})$ denotes the likelihood of the neural network estimate and $\pi_0(\Psi_{\mathrm{S}})$ is the prior distribution for the stochastic parameters of the network. As our outputs are modeled as gamma-distributed random variables, the likelihood

$\mathcal{L}(R|\Psi_D, \Psi_S)$ is given as

$$\mathscr{L}(R|\Psi_D, \Psi_S) = \prod_{i=1}^{I}\prod_{j=1}^{J} \frac{R_{i,j}^{\alpha_{NN,j}-1}\exp\left(-R_{i,j}/\beta_{NN,j}\right)}{\Gamma\left(\alpha_{NN,j}\right)\beta_{NN,j}^{\alpha_{NN,j}}}, \qquad (23)$$

where $R_{i,j}$ denotes the jth data point of the jth spectrum, $\alpha_{NN,j}$ and $\beta_{NN,j}$ are neural network outputs for the gamma distribution parameters. For the prior distribution $\pi_0(\Psi_S)$, we use an independent normal distribution $\mathcal{N}(0,1)$ for all the weights and biases of the first layer, $\pi_0(\Psi_S) \propto \prod_{p=1}^{P}\mathcal{N}\left(\Psi_{S,p};0,1\right)$ where P is the total number of parameters in the first layer and $\mathcal{N}\left(\Psi_{S,p};0,1\right)$ denotes the evaluation of the probability density at the parameter value $\Psi_{S,p}$. This particular type of Bayesian neural network is also known as a deep kernel[54] or a manifold Gaussian processes.[55] We illustrate the neural network architecture in Fig. 7.

In the log-Gaussian gamma process section, we estimate parameters of a doubly-stochastic process *via* MCMC. The Bayesian neural network architecture proposed here can be seen as an estimate of a triply-stochastic process where the neural network outputs are two stochastic process realizations $\alpha_{NN}(\nu)$ and $\beta_{NN}(\nu)$, an extension to the analytical log-Gaussian gamma process in Section 2 where the log-Gaussian parameterization of the scale process $\beta_{NN}(\nu)$ is used for mathematical convenience due to the closed form of the probability density in eqn (5).

The uncertainty quantification of the Bayesian neural network is achieved in two stages, the Bayesian convolutional layer and gamma distributed output layer. Numerical samples are generated for the parameters of the Bayesian convolutional layer during the training process. These numerical samples are propagated through the deterministic layers of the network and ultimately into the output layer. The output layer, modelled as a gamma process, outputs a gamma distribution for each prediction point which ultimately controls the uncertainty of the spectrum estimate.

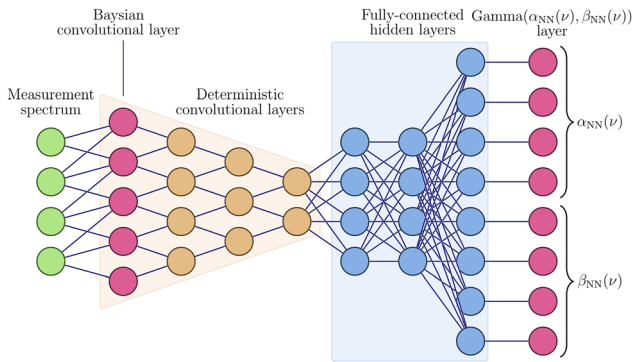

Fig. 7 A Bayesian neural network architecture for correcting spectral measurements. The first convolutional layer is modeled as a stochastic layer. A new representation of the measurement spectrum is produced *via* the convolutional layers and then decoded with the fully-connected hidden layers. The output layer is modeled as a gamma process Gamma($\alpha_{NN}(\nu)$,$\beta_{NN}(\nu)$), parameters of which are the outputs of the fully-connected hidden layers.

6 Computational details and prior distributions

We use 4 experimental Raman spectra and 4 CARS spectra to generate the synthetic training data sets. We use a wavelet-based approach[11] to obtain point estimates for the underlying Raman spectra in all 8 cases. Additionally, the method provides point estimates for the additive and multiplicative background signal which we use to estimate the parameters of the background GP model defined in eqn (18). We show the obtained Raman data point estimates for the Raman spectra and additive backgrounds in Fig. 8 and CARS point estimates for the Raman spectra and multiplicative backgrounds in Fig. 9. The four cases of measurement data are used to train their respective Bayesian neural network architectures. It should be noted that for cases with significantly different Raman spectral signatures, such as where the spectra consists of either significantly sharper or wider line shapes, the training should be done using experimental data which contain such features.

We run the DRAM algorithm with 5 proposal steps and with a length of 100 000 samples for both the log-Gaussian gamma process parameters and the GP parameters. We use a burn-in of 50 000 samples. The prior distributions for the log-Gaussian gamma process likelihood and the GP background likelihood are documented in Table 1. We use TensorFlow and TensorFlow Probability together with Keras to implement the neural network architecture.[56–58] We use the Adam optimizer for estimating the network parameters Ψ_D and Ψ_S.

The aforementioned MCMC sampling runs were done on an AMD Ryzen 3950X processor. Wall times for the MCMC sampling ranged from a couple of minutes to approximately one hour, depending on the number of data points in the spectra. The MCMC samplers can be run embarrassingly parallel for each measurement spectrum. Training the Bayesian neural network took approximately two hours on an NVIDIA 1070 graphics card for sets of 500 000 spectra. Given the small computational cost of the MCMC sampling and training the

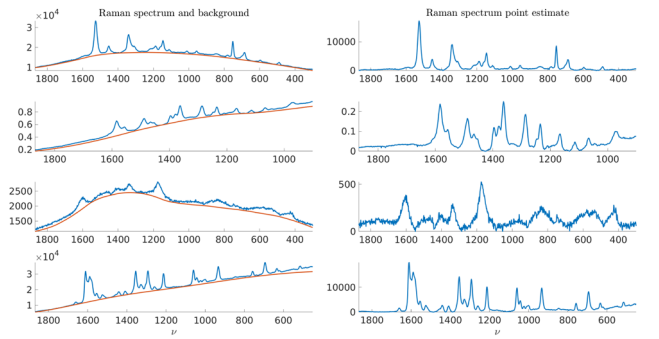

Fig. 8 On the left, experimental Raman spectra of phthalocyanine blue, naphthol red, aniline black, and red 264 pigments in blue and point estimates for their respective additive backgrounds $B(\nu)$. On the right, point estimates for the underlying Raman spectra corresponding to the Raman measurements on their left. The 4 cases are used to estimate the LGGP and GP parameters defined in the posterior distributions in eqn (10) and (18).

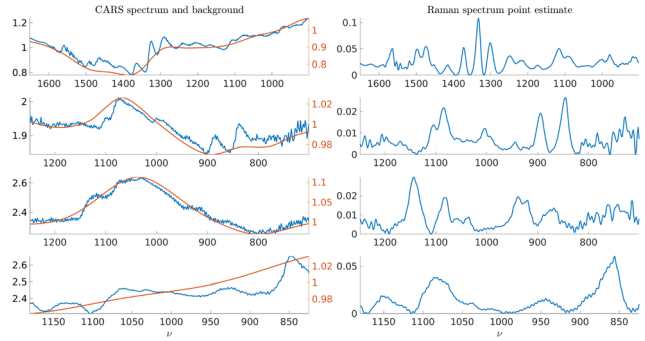

Fig. 9 On the left, experimental CARS spectra of adenosine phosphate, fructose, glucose, and sucrose in blue and point estimates for their respective multiplicative backgrounds $\varepsilon_m(\nu)$. On the right, point estimates for the underlying Raman spectra corresponding to the CARS measurements on their left. The 4 cases are used to estimate the LGGP and GP parameters defined in the posterior distributions in eqn (10) and (18).

Table 1 Prior distributions for the log-Gaussian gamma and GP parameters

Parameter	Distribution	Parameter	Distribution
$\pi_0(\alpha)$	$\mathcal{U}(0,\infty)$		
$\pi_0(c_\beta)$	$\mathcal{U}(0,\infty)$	$\pi_0(\sigma_G)$	$\mathcal{U}(1,\infty)$
$\pi_0(\mu_\beta)$	$\mathcal{U}(0,\infty)$	$\pi_0(\mu_e)$	$\mathcal{U}(0,\infty)$
$\pi_0(\sigma_{\beta,f})$	$\mathcal{U}(0,\infty)$	$\pi_0(\sigma_{e,f})$	$\mathcal{U}(0,\infty)$
$\pi_0(l_\beta)$	$\mathcal{U}(0,\infty)$	$\pi_0(l_e)$	$\mathcal{U}(0,\infty)$
$\pi_0(\sigma_\beta)$	$\mathcal{U}(0,\infty)$	$\pi_0(\sigma_e)$	$\mathcal{U}(0,\infty)$
$\pi_0(e_{\text{start}})$	$\mathcal{U}(0.90,1.10)$	$\pi_0(e_{\text{stop}})$	$\mathcal{U}(0.90,1.10)$

neural network, which are offline costs, the main computational limitation comes from loading data during inference.

7 Results

We apply the two Bayesian neural networks to 4 synthetic Raman spectra and 4 synthetic CARS spectra generated using eqn (19) and (20), respectively. The synthetic spectra were not part of the training data sets. The synthetic data and the results for Raman spectra with an additive background are shown in Fig. 10 and results for experimental Raman spectra of phthalocyanine blue, naphthol red, aniline black, and red 264 pigments are presented in Fig. 11. The experimental details of the CARS samples have been described in detail elsewhere, see for example.[37] The Raman spectra are from an online database of Raman spectra of pigments used in modern and contemporary art (the standard Pigments Checker v.5).[59]

Results for synthetic CARS spectra are shown in Fig. 12 and results for experimental CARS spectra of adenosine phosphate, fructose, glucose, and sucrose are presented in Fig. 13. The spectra themselves were not part of the training data set. The results show the median estimate of the Raman spectrum obtained from the trained Bayesian neural network along with the 90% confidence intervals of the Raman spectrum estimate. We overlay the Raman spectrum estimate with a scaled versions of the point estimates in Fig. 9. The point estimates are scaled such that the minima and maxima of the point estimate are equal to the minima and maxima of the median estimate of the

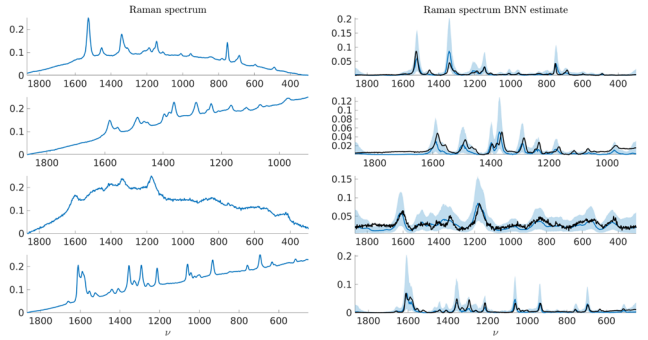

Fig. 11 On the left, experimental Raman spectra of phthalocyanine blue, aniline black, naphthol red, and red 264 pigments. On the right, corresponding Raman spectrum estimates with the median estimate shown in solid blue and the 90% confidence interval in shaded blue. The solid black line shows the ground truth spectrum.

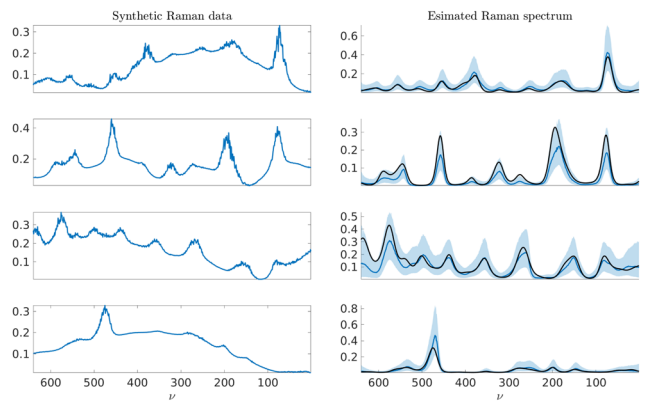

Fig. 10 On the left, synthetic Raman test spectra generated using eqn (19). On the right, corresponding Raman spectrum estimates with the median estimate shown in solid blue and the 90% confidence interval in shaded blue. The solid black line shows the ground truth spectrum.

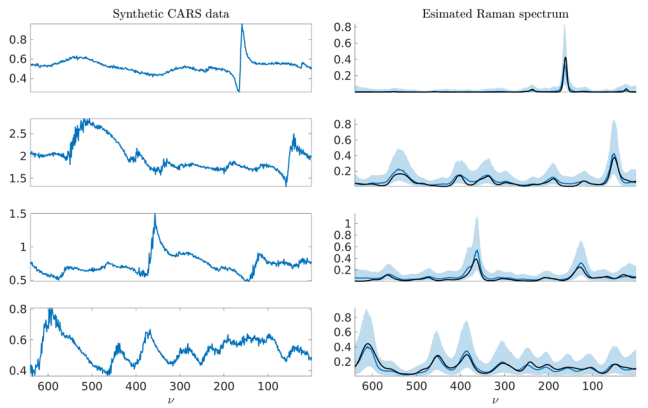

Fig. 12 On the left, synthetic CARS test spectra generated using eqn (20). On the right, corresponding Raman spectrum estimates with the median estimate shown in solid blue and the 90% confidence interval in shaded blue. The solid black line shows the ground truth spectrum.

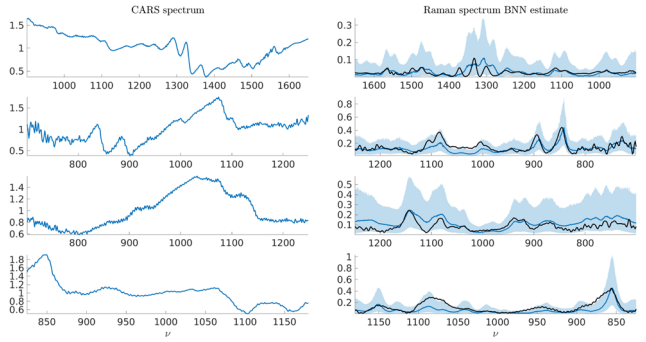

Fig. 13 On the left, experimental CARS spectra of adenosine phosphate, fructose, glucose, and sucrose. On the right, corresponding Raman spectrum estimates with the median estimate shown in solid blue and the 90% confidence interval in shaded blue. The solid black line shows the ground truth spectrum.

Raman spectrum. The results coincide with the overall shape of the point estimates, supporting the validity of the data generation approach and the Bayesian neural network design.

As additional validation for the synthetic spectrum generation approach, we compare the cost function values, see eqn (6), of the fully synthetic training spectra to the cost function values obtained for partially experimental spectra. We generate the partially experimental spectra by taking the point estimates of the experimental CARS spectra in Fig. 9 and generating sets of CARS spectra with the forward model defined in eqn (20) combined with the GP realizations. The results are in agreement which implies that our log-Gaussian gamma process is capable of generating valid Raman spectra for training neural networks. This approach is similar to approaches used for approximate Bayesian computation, see for example.[60,61] The resulting log cost function distribution of the synthetic spectra and the partially experimental spectra are provided in the ESI.†

8 Conclusions

We propose a novel approach utilizing log-Gaussian gamma processes and Gaussian processes to generate synthetic spectra and additive or multiplicative backgrounds that are statistically similar to experimental measurements, even when using a limited number of experimental spectra. The parameters of these stochastic processes are learned through Markov chain Monte Carlo methods, enabling the generation of extensive training data for neural networks by sampling from Bayesian posterior distributions of the parameters.

This data generation method is applied to train two Bayesian neural networks, specifically designed for correcting spectral measurements. One network is tailored for Raman spectra with additive backgrounds, while the other is optimized for coherent anti-Stokes Raman scattering (CARS) spectra with multiplicative backgrounds. Bayesian neural networks expand upon prior research involving neural networks for spectral corrections, offering not only point estimates but also the critical capability of uncertainty quantification.

Our approach is validated using synthetic test data generated from the stochastic processes and experimental Raman spectra of phthalocyanine blue, aniline black, naphthol red, and red 264 pigments, along with experimental CARS spectra of adenosine phosphate, fructose, glucose, and sucrose. The results demonstrate excellent agreement with deterministically obtained point estimates of the Raman spectra, while simultaneously providing valuable uncertainty estimates for the Raman spectrum estimates.

As a future avenue of research, our log-Gaussian gamma process formulation could be extended to a mixture of log-Gaussian gamma processes, similarly to mixtures of Gaussian processes.[62–64] Such an extension would allow modelling of nonstationarity and heteroscedasticity, meaning different signal or noise behaviour at different parts of measurement spectrum. A more straight-forward modification, if necessary, would be to include an additional noise term consisting of, for example, Gaussian white noise process to the log-Gaussian gamma process formulation.

In addition, log-Gaussian gamma processes might be used to model other inherently positive measurements such as reflectance, absorbance, fluorescence, or transmittance spectra[65–67] and other non-spectroscopic data sets like pollutant or protein concentrations or masses of stellar objects, for which Gaussian processes have been used[68–70] Modifications to the log-Gaussian gamma process and the Bayesian neural network should be minimal due to the uninformative priors and general purpose structure of them. The generation of synthetic data sets for other types of problems requires an appropriate forward model. If the forward model has non-local behaviour, it might warrant architectural changes to the Bayesian neural network such as augmenting the first layer with a dense layer for modelling the non-local dependencies in the data.

A comparison study similar to[35] would be an interesting avenue of future research. However, a direct comparison between non-Bayesian and Bayesian neural networks is not straight-forward due to the missing uncertainty quantification of the non-Bayesian neural network estimates. One could possibly use the Dropout regularization as an approximate Bayesian approach.[71] This could be combined with simulation-based calibration[72] to provide an appropriate one-to-one comparison between the non-Bayesian and Bayesian neural networks.

Author contributions

T. H.: conceptualization, formal analysis, investigation, methodology, software, visualization. E. M. V. and L. L: resources. M. T. M.: investigation, methodology. L. R.: funding acquisition, project administration, supervision. All authors: writing – original draft, writing – review & editing.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

The authors were supported by Research Council of Finland (grant number 353095). T. H. thanks Ilmari Vahteristo for the fruitful discussions.

Notes and references

1 S. P. Mulvaney and C. D. Keating, *Anal. Chem.*, 2000, **72**, 145–158.
2 C. Krafft, B. Dietzek, J. Popp and M. Schmitt, *J. Biomed. Opt.*, 2012, **17**, 040801.
3 J. P. R. Day, K. F. Domke, G. Rago, H. Kano, H.-O. Hamaguchi, E. M. Vartiainen and M. Bonn, *J. Phys. Chem. B*, 2011, **115**, 7713–7725.
4 H. F. M. Boelens, P. H. C. Eilers and T. Hankemeier, *Anal. Chem.*, 2005, **77**, 7998–8007.
5 S. He, W. Zhang, L. Liu, Y. Huang, J. He, W. Xie, P. Wu and C. Du, *Anal. Methods*, 2014, **6**, 4402–4407.
6 F. Gan, G. Ruan and J. Mo, *Chemom. Intell. Lab. Syst.*, 2006, **82**, 59–65.
7 C. M. Galloway, E. C. L. Ru and P. G. Etchegoin, *Appl. Spectrosc.*, 2009, **63**, 1370–1376.
8 E. M. Vartiainen, H. A. Rinia, M. Müller and M. Bonn, *Opt. Express*, 2006, **14**, 3622–3630.
9 Y. Kan, L. Lensu, G. Hehl, A. Volkmer and E. M. Vartiainen, *Opt. Express*, 2016, **24**, 11905–11916.
10 M. Chi, X. Han, Y. Xu, Y. Wang, F. Shu, W. Zhou and Y. Wu, *Appl. Spectrosc.*, 2019, **73**, 78–87.
11 T. Härkönen and E. Vartiainen, *Opt. Continuum*, 2023, **2**, 1068–1076.
12 K. H. Liland, T. Almøy and B.-H. Mevik, *Appl. Spectrosc.*, 2010, **64**, 1007–1016.
13 J. A. Weyn, D. R. Durran and R. Caruana, *J. Adv. Model. Earth Syst.*, 2020, **12**, e2020MS002109.
14 J. Ritvanen, B. Harnist, M. Aldana, T. Mäkinen and S. Pulkkinen, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 2023, **16**, 1654–1667.
15 A. M. Abdalla, I. H. Ghaith and A. A. Tamimi, 2021 International Conference on Information Technology (ICIT), 2021, pp. 622-626.
16 S. J. Hamilton and A. Hauptmann, *IEEE Trans. Med. Imaging*, 2018, **37**, 2367–2377.
17 C. B. Monti, M. Codari, M. van Assen, C. N. De Cecco and R. Vliegenthart, *J. Thorac. Imaging*, 2020, **35**, S58–S65.
18 S. Suganyadevi, V. Seethalakshmi and K. Balasamy, *Int. J. Multimed. Inf. Retr.*, 2022, **11**, 19–38.
19 P. Sharma and A. Singh, 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2017, pp. 1-5.
20 P. P. Shinde and S. Shah, 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018, pp. 1-6.
21 W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders and K.-R. Müller, *Proc. IEEE*, 2021, **109**, 247–278.
22 L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie and L. Farhan, *J. Big Data*, 2021, **8**, 53.
23 Z. Li, F. Liu, W. Yang, S. Peng and J. Zhou, *IEEE Trans. Neural Netw. Learn. Syst.*, 2022, **33**, 6999–7019.
24 J. Liu, M. Osadchy, L. Ashton, M. Foster, C. J. Solomon and S. J. Gibson, *Analyst*, 2017, **142**, 4067–4074.
25 J. Wahl, M. Sjödahl and K. Ramser, *Appl. Spectrosc.*, 2020, **74**, 427–438.
26 M. T. Gebrekidan, C. Knipfer and A. S. Braeuer, *J. Raman Spectrosc.*, 2021, **52**, 723–736.
27 M. Kazemzadeh, C. L. Hisey, K. Zargar-Shoshtari, W. Xu and N. G. Broderick, *Opt. Commun.*, 2022, **510**, 127977.
28 R. Luo, J. Popp and T. Bocklitz, *Analytica*, 2022, **3**, 287–301.
29 C. M. Valensise, A. Giuseppi, F. Vernuccio, A. De la Cadena, G. Cerullo and D. Polli, *APL Photonics*, 2020, **5**, 061305.
30 R. Houhou, P. Barman, M. Schmitt, T. Meyer, J. Popp and T. Bocklitz, *Opt. Express*, 2020, **28**, 21002–21024.
31 Z. Wang, K. O' Dwyer, R. Muddiman, T. Ward, C. H. Camp Jr. and B. M. Hennelly, *J. Raman Spectrosc.*, 2022, **53**, 1081–1093.
32 R. Junjuri, A. Saghi, L. Lensu and E. M. Vartiainen, *Opt. Continuum*, 2022, **1**, 1324–1339.
33 A. Saghi, R. Junjuri, L. Lensu and E. M. Vartiainen, *Opt. Continuum*, 2022, **1**, 2360–2373.
34 R. Junjuri, A. Saghi, L. Lensu and E. M. Vartiainen, *RSC Adv.*, 2022, **12**, 28755–28766.
35 R. Junjuri, A. Saghi, L. Lensu and E. M. Vartiainen, *Phys. Chem. Chem. Phys.*, 2023, **25**, 16340–16353.
36 M. T. Moores, K. Gracie, J. Carson, K. Faulds, D. Graham and M. Girolami, *arXiv*, 1604.07299, 2016.
37 T. Härkönen, L. Roininen, M. T. Moores and E. M. Vartiainen, *J. Phys. Chem. B*, 2020, **124**, 7005–7012.
38 X. Wang, F. Guo, K. A. Heller and D. B. Dunson, *Adv. Neural Inf. Process Syst.*, 2015, 451–459.
39 M. Magris and A. Iosifidis, *Artif. Intell. Rev.*, 2023, 1–51.
40 M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov and S. Nahavandi, *Inf. Fusion*, 2021, **76**, 243–297.
41 T. Cui, K. J. Law and Y. M. Marzouk, *J. Comput. Phys.*, 2016, **304**, 109–137.
42 M. Morzfeld, X. Tong and Y. Marzouk, *J. Comput. Phys.*, 2019, **380**, 1–28.
43 A. X. Yang, M. Robeyns, X. Wang and L. Aitchison, *arXiv*, 2308.13111, 2023.
44 M. Sharma, S. Farquhar, E. Nalisnick and T. Rainforth, *arXiv*, 2211.06291, 2023.
45 D. Antonio, H. O'Toole, R. Carney, A. Kulkarni and A. Palazoglu, *arXiv*, 2306.16621, 2023.
46 M. H. Mozaffari and L.-L. Tay, 2021 5th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI), 2021, pp. 1-6.
47 C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2005.
48 T. Härkönen, E. Hannula, M. T. Moores, E. M. Vartiainen and L. Roininen, *Foundations of Data Science*, 2023, DOI: 10.3934/fods.2023008.

49 J. Møller, A. R. Syversveen and R. P. Waagepetersen, *Scand. J. Stat.*, 1998, **25**, 451–482.
50 J. M. Hilbe, *Negative Binomial Regression*, Cambridge University Press, 2nd edn, 2011.
51 M. Teng, F. Nathoo and T. D. Johnson, *J. Stat. Comput. Simul.*, 2017, **87**, 2227–2252.
52 H. Haario, M. Laine, A. Mira and E. Saksman, *Stat. Comput.*, 2006, **16**, 339–354.
53 N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, *J. Mach. Learn. Res.*, 2014, **15**, 1929–1958.
54 A. G. Wilson, Z. Hu, R. Salakhutdinov and E. P. Xing, Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, Cadiz, Spain, 2016, pp. 370-378.
55 R. Calandra, J. Peters, C. E. Rasmussen and M. P. Deisenroth, 2016 International joint conference on neural networks (IJCNN), 2016, pp. 3338-3345.
56 M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu and X. Zheng, *arXiv*, 1603.04467, 2015.
57 J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman and R. A. Saurous, *arXiv*, 1711.10604, 2017.
58 F. Chollet *et al.*, *Keras*, 2015, **https://keras.io**.
59 The standard Pigments Checker v.5, **https://chsopensource.org/pigments-checker/**, Accessed: 2023-02-14.
60 B. Jiang, T.-Y. Wu, C. Zheng and W. H. Wong, *Stat. Sin.*, 2017, **27**, 1595–1618.
61 C. Grazian and Y. Fan, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2020, **12**, e1486.
62 V. Tresp, *Adv. Neural Inf. Process Syst.*, 2001, 654–660.
63 C. E. Rasmussen and Z. Ghahramani, *Advances in Neural Information Processing Systems 14*, MIT Press, 2002, pp.881–888.
64 T. Härkönen, S. Wade, K. Law and L. Roininen, Mixtures of Gaussian Process Experts with SMC^2, *arXiv*, 2208.12830, 2022.
65 S. Van Wittenberghe, J. Verrelst, J. P. Rivera, L. Alonso, J. Moreno and R. Samson, *J. Photochem. Photobiol., B*, 2014, **134**, 37–48.
66 M. Lázaro-Gredilla, M. K. Titsias, J. Verrelst and G. Camps-Valls, *IEEE Geosci. Remote. Sens.*, 2014, **11**, 838–842.
67 M. Ghosh, Y. Li, L. Zeng, Z. Zhang and Q. Zhou, *IISE Trans.*, 2021, **53**, 787–798.
68 N. Lawrence, G. Sanguinetti and M. Rattray, *Adv. Neural Inf. Process Syst.*, 2006, 785–792.
69 Y. Bu, Y. B. Kumar, J. Xie, J. Pan, G. Zhao and Y. Wu, *Astrophys. J., Suppl. Ser.*, 2020, **249**, 7.
70 T. Härkönen, A.-M. Sundström, J. Tamminen, J. Hakkarainen, E. Vakkilainen and H. Haario, *Int. J. Uncertain. Quantif.*, 2023, **13**, 41–59.
71 Y. Gal and Z. Ghahramani, Proceedings of The 33rd International Conference on Machine Learning, New York, New York, USA, 2016, pp. 1050-1059.
72 S. Talts, M. Betancourt, D. Simpson, A. Vehtari and A. Gelman, *arXiv*, 1804.06788, 2020.