Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **508,** 2090–2097 (2021)
Advance Access publication 2021 September 20

https://doi.org/10.1093/mnras/stab2623

Density estimation with Gaussian processes for gravitational wave posteriors

V. D'Emilio ,* R. Green and V. Raymond*
Cardiff University, Cardiff CF24 3AA, UK

Accepted 2021 September 10. Received 2021 September 10; in original form 2021 April 12

ABSTRACT

The properties of black hole and neutron-star binaries are extracted from gravitational waves (GW) signals using Bayesian inference. This involves evaluating a multidimensional posterior probability function with stochastic sampling. The marginal probability distributions of the samples are sometimes interpolated with methods such as kernel density estimators. Since most post-processing analysis within the field is based on these parameter estimation products, interpolation accuracy of the marginals is essential. In this work, we propose a new method combining histograms and Gaussian processes (GPs) as an alternative technique to fit arbitrary combinations of samples from the source parameters. This method comes with several advantages such as flexible interpolation of non-Gaussian correlations, Bayesian estimate of uncertainty, and efficient resampling with Hamiltonian Monte Carlo.

Key words: gravitational waves – methods: data analysis.

1 INTRODUCTION

The first detection of gravitational waves (GW) in 2015 (Abbott et al. 2016a) sparked a new era of Astronomy. Several years on from that event the number of detected GWs keeps increasing and within this decade we expect to observe $O(10^3)$ signals (Abbott et al. 2020b) from compact binary coalescences (CBCs). This huge progress brings with it the challenge of efficiently analysing a large number of events. To address these computational challenges, machine-learning techniques are being increasingly investigated within the field of GW physics (Cuoco et al. 2020). Many studies have focused on speeding parameter estimation of the source parameters of the signals with various techniques, such as deep learning (George & Huerta 2018), variational autoencoders (Gabbard et al. 2019), and autoregressive neural flows (Green, Simpson & Gair 2020). Other work has focused on combining detection and parameter estimation with deep neural networks (Fan et al. 2019) as well as using neural networks to rapidly generate surrogate waveforms (Chua, Galley & Vallisneri 2019; Khan & Green 2020).

While the research efforts to speed up or completely revolutionize parameter estimation are ongoing, the issue of how to effectively deal with a large number of results from different events remains. In particular, how to streamline the analysis of the results, while maintaining accuracy. In this work, we demonstrate the efficiency and usefulness of using Gaussian processes (GP) for post-processing parameter estimation results of CBCs. Applications of GPs in the field of GWs span a wide range of use-cases, such as marginalizing waveform errors (Moore et al. 2016), regression of analytical waveforms (Setyawati, Pürrer & Ohme 2020), predictions of population synthesis simulations (Barrett et al. 2016), hierarchical population inference (Taylor & Gerosa 2018), and Equation of State (EOS) calculations (Landry & Essick 2019). They have also been exploited for the development of fast parameter estimation with RIFT sampler (Lange, O'Shaughnessy & Rizzo 2018).

Here, we exploit GPs to estimate probability density functions (PDFs) from parameter estimation of GW signals. Non-parametric density estimation from a finite set of samples is an active research field in machine learning and statistics (Murray, MacKay & Adams 2008; Papamakarios, Pavlakou & Murray 2017; Wang & Scott 2019).

For most GW analysis, histograms are usually the preferred estimators to visualize the marginal posterior PDFs and to avoid oversmoothing sharp features, but often are not convenient for post-processing analyses such as population inference. These sorts of analyses either reweight the posterior samples directly (Abbott et al. 2020a) or need to estimate a continuous representation of the GW posterior density surface. Several density estimation methods such as Dirichlet processes (Del Pozzo et al. 2018), Gaussian mixture models (Talbot & Thrane 2020), and others have been employed to address this problem specifically for GWs. As well as these, A closely related method to GPs (Kanagawa et al. 2018), Gaussian kernel density estimators (KDEs) are sometimes employed in GW analyses (Lynch et al. 2017; Pitkin, Messenger & Fan 2018; Pang et al. 2020)

These KDEs are often effective but they assume correlations between parameters to be linear and smooth, making this method sometimes limited in flexibility. There exist many variations of the KDE algorithm to take into account specific interpolations problems, but there is not a single implementation that is guaranteed to be robust against all possibilities. A specific KDE implementation might solve an issue in one case and be the cause of some inaccuracies in another (Wand & Jones 1994).

We implement a single technique that can interpolate arbitrary multidimensional slices in parameter space, which can handle both

* E-mail: DemilioV@cardiff.ac.uk (VD'E); RaymondV@cardiff.ac.uk (VR)

simple and difficult space morphology, such as sharp bounds and non-Gaussian correlations. Our modelling tool is based on the histogram density estimate, combining the histogram's accurate treatment of the samples' features with the predictive capabilities of GPs. An additional advantage of this technique is that it can provide a Bayesian measure of uncertainty from the finite (and sometimes small) number of samples for post-processing analysis. This measure of model uncertainty could then be incorporated into any analysis where the marginalized posterior density is used.

In Section 2, we describe our density estimation technique in the context of GW parameter estimation and machine learning. We propose a series of example applications in Section 3, which allows us to discuss the advantageous features of our method. Finally, in Section 4, we summarize our findings and suggest future extensions of this work.

2 METHODS

In this section, we introduce the mathematical framework of the techniques discussed. In Section 2.1, we discuss the Bayesian inference problem for GWs and the density estimation techniques currently employed in the field. In Section 2.2, we outline the fundamentals of GPs and their interpretation for interpolating a posterior density surface. We then describe the details of our GP implementation and how to model probability densities from parameter estimation.

2.1 Bayesian inference and density estimators

We describe the GW data in the detector d as the sum of a waveform model $h(\boldsymbol{\theta})$ and a combination of instrumental noise, which we assume to be Gaussian. The probability of observing data parametrized by $\boldsymbol{\theta} = \{\theta_1, \theta_2, ..., \theta_N\}$, can be defined as the probability that $r = d - h$ is the realization of the instrumental noise. This likelihood can be written as

$$p(\mathrm{d}|\boldsymbol{\theta}) \propto \exp\left(-\frac{1}{2}\langle \mathrm{d} - h(\boldsymbol{\theta})|\mathrm{d} - h(\boldsymbol{\theta})\rangle\right), \tag{1}$$

where $\langle a|b\rangle$ denotes the inner product between two waveforms a and b and is defined as (Cutler & Flanagan 1994)

$$\langle a|b\rangle = 4\mathrm{Re}\int_0^\infty \frac{\tilde{a}(f)\tilde{b}^*(f)}{S_n(f)}\mathrm{d}f, \tag{2}$$

where $S_n(f)$ is the one-sided power spectral density (PSD) and $\tilde{a}$ denotes the Fourier transform of the gravitational waveform a.

By choosing astrophysically motivated priors over the model parameters, we can use the Bayesian framework to calculate the posterior probability distribution for the source parameters (Thrane & Talbot 2019)

$$\begin{aligned} p(\boldsymbol{\theta}|\mathrm{d}) &= \frac{p(\mathrm{d}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathrm{d})} \\ &\propto p(\mathrm{d}|\boldsymbol{\theta})p(\boldsymbol{\theta}). \end{aligned} \tag{3}$$

The posterior probability is generally a 15-dimensional surface for a circular binary black hole (BBH) merger but can be 17-dimensional in the case of a binary neutron star (BNS) merger. The dimensionality depends on the physical parameters describing the signals. Generally, these are distinguished between extrinsic parameters, such as sky localization, and intrinsic parameters, such as the masses of the sources.

Posterior distributions for specific parameters can then be found by marginalizing over all other parameters,

$$p(\theta_i|\mathrm{d}) = \int p(\vec{\theta}|\mathrm{d})\mathrm{d}\theta_1...\mathrm{d}\theta_{i-1}\mathrm{d}\theta_{i+1}...\mathrm{d}\theta_N. \tag{4}$$

The posterior is, however, generally intractable and therefore must be evaluated via stochastic methods such as Markov chain Monte Carlo (MCMC) and nested sampling, these are implemented (and specifically tuned for the GW problem) in Bayesian inference packages such as LALInference (Veitch et al. 2015) and `bilby` (Ashton et al. 2019).

2.2 Density estimation with Gaussian processes

2.2.1 Definition and interpretation

GPs are interpolation methods with a probabilistic interpretation, they are built on a Bayesian philosophy, which allows you to update your beliefs based on new observations. The process can be understood as an infinite-dimensional generalization of multivariate normal distributions, such that any finite collection of points within the domain of the process are related by a multivariate Gaussian distribution. As data is observed, the GP is *conditioned* and the range of possible functions that can explain the observations is constrained. As such a GP is defined by a mean, which represents the expectation value for the best-fitting function, and by a covariance matrix, called a kernel, which measures the correlations between observations (Williams & Rasmussen 2006). In the absence of observations, the GP predictions will revert to a prior mean function, which is usually chosen to be zero, and which properties are determined by the kernel architecture. Mathematically this is written as

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), \kappa(\boldsymbol{x}, \boldsymbol{x}')), \tag{5}$$

where the mean and covariance are denoted as

$$m(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x})],$$

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}\left[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x}') - m(\boldsymbol{x}'))\right].$$

We can then model a surface y conditioned on our observations as

$$y_*|f, x \sim \mathcal{N}\left(m(x_*), \sigma_*^2\right), \tag{6}$$

where, in this application, the dimensionality of f will depend on how many parameters $p(\theta_i|d)$ has been marginalized over.

The non-parametric nature of GPs makes this technique flexible, but it can be computationally expensive as the whole training set needs to be taken into account at each prediction. The standard implementation has $\mathcal{O}(N^3)$ computations and $\mathcal{O}(N^2)$ storage, this then becomes prohibitive for ~10k data observations or more. To tackle this issue, it is common to use sparse inference methods, which approximate the conditioning of the GP over a set of $M << N$ 'inducing' points. The evaluation over the inducing points M is then much cheaper than for an 'exact' GP resulting in $\mathcal{O}(NM^2)$ computations rather than $\mathcal{O}(N^3)$ (Quiñonero-Candela & Rasmussen 2005; Hensman, Fusi & Lawrence 2013). As well as sparse methods one can exploit multi-GPU parallelization and methods like linear conjugate gradients to distribute the kernel matrix evaluations which then allows for exact inference to be performed on a short time-scale (Wang et al. 2019). In this work, however, we find that sparse approximations were accurate enough to effectively model the marginalized posterior surfaces that we were interested in. Moreover, once a GP has been 'trained' over the data, it is possible to draw infinitely many function realizations from it without recomputing the expensive covariance matrix.

A recognized advantage of GPs is reliable uncertainty estimate when making predictions over unseen data. In this application, we are not interested in predicting the value of the posterior in unexplored regions of the parameter space, but only in generating a faithful model where we have posterior samples. In regions within the space of parameters, the GP variance depends on our choice of training points, which is useful to assess the accuracy of our density estimation. In terms of uncertainty estimation, this can be explained as our model having very low *epistemic* uncertainty everywhere, we then seek to estimate the *aleatoric* uncertainty due to our model fit around the random fluctuations in the histogram densities which are used to train the GP.

2.2.2 Model construction

In this application, we want to use a GP to estimate the marginalized posterior density for any subset of parameters. We train our GP using the normalized histogram counts over a grid of points, i.e. the centroids of the histogram bins, that cover the marginalized parameter space. We then fit our GP to this discrete set of points to generate a continuous representation of the surface.

An important choice when modeling a system using GPs is the choice of kernel, this encodes your assumptions about the relationship or covariance between data points. in this paper we used a combination of the RBF and Matern ($\frac{1}{2}$ or $\frac{5}{2}$) kernels. In the case of periodic parameters (such as the sky location), the periodic version of the chosen kernel (MacKay et al. 1998) may be necessary. To account for exceptionally non-trivial correlations between parameters, a non-stationary kernel, such as deep kernels (Wilson et al. 2016) can be used. Further technical details regarding this choice and our data pre-processing scheme (which also had a significant impact on our model accuracy) are included in Appendix A.

We employ `TensorFlow` and `GPFlow` to implement our GP training infrastructure, which includes two inference schemes: exact inference for one to two dimensional problems ($\mathcal{O} \sim 1000$ samples) and sparse inference for higher dimensionality due to computational costs. As well as a difference in the inference scheme, when extending this method to higher dimensions, our choice of training data changes. When creating the grid over four dimensions, due to the sparsity of the parameter space, we find that the typical set has a volume of $O(1$ per cent) relative to the total prior volume [this is a common problem associated with the curse of dimensionality (Betancourt 2017)]. We, therefore, choose to discard the empty bins and encode our knowledge of these points through the choice of prior over our GP.

Since the model is constructed with converged posterior samples, there is no probability support where the histogram bins are empty. To encode this, we set the mean of the GP to be equal to zero, such that far away from the training data the model will have a high variance but a mean of zero.

To estimate the density for a given region of parameter space, we then simply evaluate the GP at those parameters, i.e.

$$\begin{aligned} p(\vec{\theta} = \vec{x_*}|d) &\approx y_*|f, x \\ &\sim \mathcal{N}(f(\vec{x_*}), \sigma_*^2). \end{aligned} \tag{7}$$

The choice to set the GP prior to zero means that we would be allowing for negative probaility densities, to avoid this we apply the ReLU function (Nair & Hinton 2010) as a layer on top of the density evaluation. This sets all negative values to zero meaning that some points in parameter space will be distributed as a truncated-Gaussian.

Due to bounded priors (e.g at mass ratio $m_2/m_1 := q = 1$), the posterior surface often presents sharp discontinuities and therefore the surface is only *piece-wise continuous*. GPs are in principle flexible enough to model any surface including piece-wise continuous ones, however, we found in practice that it is more favourable to decompose our density function into two components, one smooth, continuous function, and one step function. We do this by multiplying the density and our GP estimate by a step function, which is zero at any discontinuities and 1 otherwise.

$$\pi(\vec{x_*}) = \begin{cases} 1 & \text{if } x_{\min} < \vec{x_*} < x_{\max} \\ 0 & \text{otherwise} \end{cases}.$$

Multiplying by this step function is then analogous to imposing a prior over our posterior surface, i.e. it allows us to rewrite the equation (7) as

$$\begin{aligned} p(\vec{\theta} = \vec{x_*}|\mathrm{d})\pi(\vec{x_*}) &\approx (y_*|f, x)\pi(\vec{x_*}) \\ p(\vec{\theta} = \vec{x_*}|\mathrm{d}) &\sim \mathcal{N}(f(\vec{x_*}), \sigma_*^2)\pi(\vec{x_*}) \end{aligned}. \tag{8}$$

We are free to encode our knowledge in this way and perform the decomposition as we do not change the original posterior surface that we would like to model in any way. This enhances the robustness of the model against all discontinuities, including artificial cuts in parameter space that might be required for post-processing analysis.

The variance of the GP depends on the kernel, but also on the noise variance parameter of the likelihood. Usually, the noise variance is given by a single number, i.e. homoskedastic noise, which reflects the random fluctuations of the posterior samples. In low-dimensional examples, where we employ an exact inference scheme, we can assign multiple values to the noise variance, i.e. heteroskedastic noise (McHutchon & Rasmussen 2011). In such instances, we are then able to propagate the error from the histogram on the density estimate, which is simply given by the Poisson noise in each bin $\sigma_{\mathrm{bin}} \sim \sqrt{N_{\mathrm{counts}}}$. Incorporating heteroskedastic errors within a sparse inference scheme is an area of current research in the field of machine learning (Liu, Ong & Cai 2020).

It is common practice to build an interpolation of a posterior surface in order to draw more samples from it. As our model is implemented in `TensorFlow,` we can quickly draw more samples from the marginalized posteriors using the many samplers available in the package library, such as Hamiltonian Monte Carlo (HMC) (Betancourt 2017).

3 RESULTS

In this section, we present our model and a series of example applications for GWs. In Section 3.1, we illustrate the method on a simple one-dimensional analytical example. In Section 3.2, we show examples of common post-processing applications for our density estimation tool. Finally, we discuss our treatment of GP model uncertainty and how we propagate it to produce uncertainty on the marginalized posterior distributions.

3.1 Analytical 1D example

Our proposed GP modelling technique is by construction flexible and robust against all distribution morphologies. To illustrate this, we construct a non-trivial one-dimensional example: an inverse gamma function $f(x, \alpha) = \frac{x^{-\alpha-1}}{\Gamma(a)}\exp(-\frac{1}{x})$, with $\alpha = 2$ and a sharp bound at $x = 0.75$.

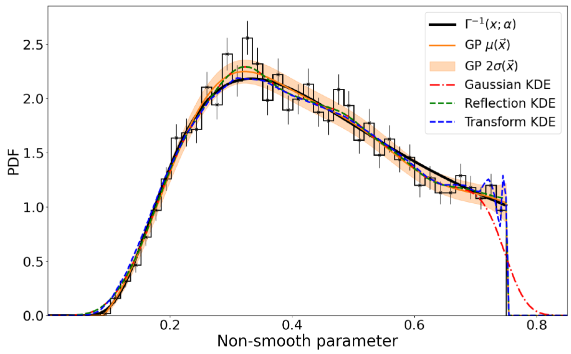

Figure 1. Interpolation of a bounded one-dimensional inverse gamma density function (in solid black) with our GP-based method (in solid orange). The histogram points used to generate the model and its uncertainty are shown as black points with error bars. Alternative KDE methods are shown for comparison as coloured dashed lines.

In Fig. 1, we show our GP model mean prediction and uncertainty, compared to a Gaussian KDE from `scipy.stats` (Virtanen et al. 2020) and two KDE transformations implemented in `PESummary` (Hoy & Raymond 2020), a commonly used post-processing package in GW astronomy. The *reflection* and *transform* KDEs, are examples of augmentations on the standard (Gaussian) KDE, and are generally used to model difficult features introduced at the boundaries of posterior distributions. Both of these improvements to the standard KDE apply a transformation at the boundary which implicitly assumes some distributional features (see Hoy & Raymond 2020 for more details). A Gaussian Process, on the other hand, makes no assumptions about the distributional shape and can in principle fit any distribution.

We show an example in Fig. 1 where our GP is able to well model the posterior and the *reflection* KDE provides a better fit than the other KDE methods. The *transform* KDE is more sensitive to noisy features in the samples and can present artefacts, while the Gaussian KDE oversmooths the sharp cut at 0.75. Following this illustrative example, there are others where the *reflection* KDE is less appropriate. This example was chosen to highlight a case where the choice of KDE is important to fit the distribution well. While synthetic and not representative, it does illustrate features that can and do happen in GW astronomy when analysing posteriors. In examples such as this our GP model provides an alternative method to KDEs, requires less hand-tuning, and also provides a Bayesian estimate of the error on the density estimate, as propagated from the histogram errors.

3.2 GW applications

We now look at a few important post-processing problems in GW astrophysics. The training time required to generate the models presented in this section is of the order of ~2 min, with variations due to the dimensionality of the surface and to the inference scheme employed. To assess the quality of the model in more than one dimension, we decide to resample the surrogate surface and compare the new samples to the original set, part of which has been used for training. All samples used in the following sections are taken from the Bilby GWTC-1 catalogue (Romero-Shaw et al. 2020a).

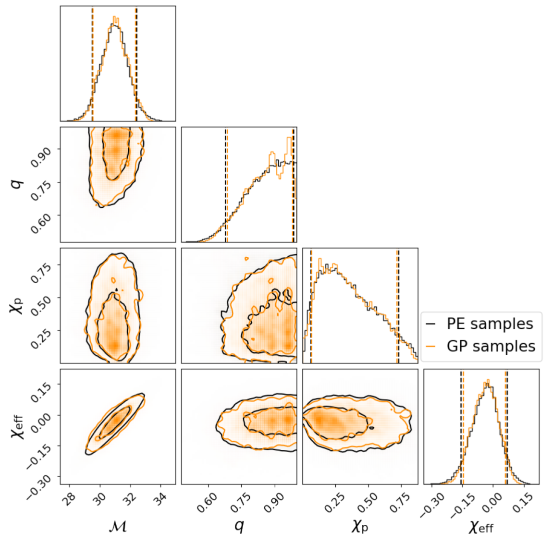

Figure 2. Corner plot of the intrinsic parameters of GW150914, drawn from our GP surrogate (in orange) compared to the original PE samples (in black).

Table 1. Source properties of the intrinsic parameters of GW150914, original samples, and samples from the GP interpolation.

	GP samples	PE samples
Chirp mass $\mathcal{M}/\mathrm{M}_\odot$	$30.95^{+0.93}_{-0.97}$	$30.96^{+0.86}_{-0.89}$
Mass ratio q	$0.87^{+0.09}_{-0.12}$	$0.87^{+0.09}_{-0.12}$
Effective precession Spin component χ_p	$0.33^{+0.26}_{-0.19}$	$0.32^{+0.27}_{-0.19}$
Effective inspiral Spin component χ_eff	$-0.04^{+0.07}_{-0.07}$	$-0.04^{+0.06}_{-0.07}$

3.2.1 *Catalogue of GW properties*

GW detection parameters can be distinguished between those intrinsic to the sources, such as the component masses, and those extrinsic to them, such as the sky location. Interpolating the marginal posteriors of combinations of these parameters is often necessary for post-processing. The following example illustrates a simple case where one can use a GP to interpolate the intrinsic parameters for a given detection. In practice, this could then be repeated for entire GW catalogues so that these interpolated posterior surfaces are then combined for population inferences on the sources of GWs.

For this example, we interpolate the marginal posterior distribution of the intrinsic parameters of the first BBH detection GW150914 (Abbott et al. 2016b), parametrized as follows: chirp mass $\mathcal{M}$, mass ratio $q = m_2/m_1$ (where $m_1 > m_2$), effective inspiral spin component χ_eff, and effective precession spin $\vec{\chi}_\mathrm{p}$, defined by the spin components that lie in the orbital plane (Schmidt, Ohme & Hannam 2015). In Fig. 2, we compare the marginal distributions sampled from our GP model to the original PE samples. We can visually assess that the correlations between parameters are accurately reconstructed as the 50 and 90 per cent contour lines overlap for each pair of parameters. In Table 1, we report the mean and 90 per cent confidence intervals of the samples drawn from our model and which we find in agreement to the values from the original samples within the expected uncertainty.

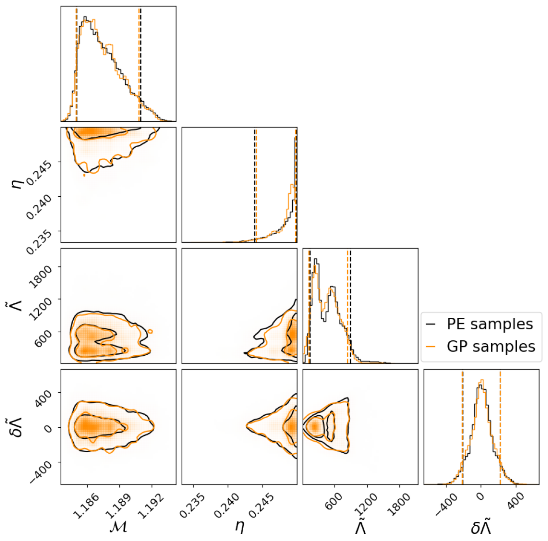

Figure 3. Corner plot of the mass and tidal parameters of GW170817, drawn from our GP model (in orange), compared to original PE samples in black.

3.2.2 *Accurate interpolation for conditional integrals*

Many astrophysical inquiries in GW astronomy require evaluating conditional integrals across parameter space, which, in turn, require sampling additional posterior points constrained to a hyperplane. This is, for instance, the case when estimating the EOS from BNS collisions, an important post-processing analysis that allows us to probe extreme conditions of matter (Abbott et al. 2018). This is possible because the compactness of the objects is imprinted in the gravitational waveform and can be measured by the tidal deformability parameters. The EOS integral involves evaluating the marginal posterior distribution over the masses ($\mathcal{M}$, η) and tidal parameters ($\tilde{\Lambda}$, $\delta\tilde{\Lambda}$), subject to constraints between those parameters as parametrized by the EOS.

There are instances where the marginal posterior for these parameters contain non-linear correlations, as is the case for the first BNS event GW170817 (Abbott et al. 2017a). We test our interpolation model over this four-dimensional surface. In Fig. 3, we compare the marginal distributions sampled from our GP model to the original PE samples. We see that our GP is able to faithfully represent the marginalized posterior surface, in particular, we see that there is good agreement between the 90 per cent credible intervals. When looking at the 2D contours see that the 50 and 90 per cent levels agree very well and that the GP model is able to capture degenerate features and bi-modalities. Finally, our interpolation of the surface can be resampled efficiently and for this example, we obtained 750k samples in ~5 min, (depending on hardware) using an HMC sampler. Hence, this method can be advantageous over traditional methods, where the interpolation is generally performed with a Gaussian KDE by transforming the symmetric mass ratio parameter to be $log(0.25 - \eta)$ (Pang et al. 2020) and there is no measure of uncertainty over the fit.

3.2.3 *Propagating GP uncertainty*

GPs provide a fully Bayesian estimation of the uncertainty over model predictions, as the full covariance matrix between posterior samples is computed. In each of the GW applications shown so far we have utilized the mean prediction of the GP function. This uncertainty measurement can be very important in many cases, however, here we illustrate with a single example how one can extract the uncertainty from the modelling. Accurate localization of a gravitational signal can be of fundamental importance for multimessenger astronomy (Grover et al. 2014; Abbott et al. 2017b) and for measurements of cosmological parameters with dark sirens (Soares-Santos et al. 2019). As the localization accuracy decreases, the marginal posteriors for the sky location parameters can look degenerate and non-Gaussian. We build an interpolation of the sky location parameters, right ascension (RA) and declination (Dec.), of GW150914.[1] This event was observed by only two detectors, so despite its high SNR, its sky location presents a typical ring-like shape.

The sky localization parameter space contains several interesting features, such as the highly curved correlation which are in principle difficult to model. For this particular example, the simple kernels used throughout the paper were sufficient and used here for simplicity. Note that, in general, we formally encode periodic parameters such as *ra* using a periodic version of the chosen kernel (MacKay et al. 1998) (see A2).

The uncertainty measure produced by the GP is a Gaussian distribution about any given point on the surface, when considering the entire surface the combination of these Gaussians can be interpreted as a range of plausible density surfaces for any given confidence level (e.g. 2σ). The uncertainty on the 1D marginal distributions can then be obtained from an upper and lower bound for each point in the surface (given by the GP error σ, equation 6) and then marginalizing these across one of the dimensions to obtain an uncertainty estimate about the mean 1D predicted posterior density. For brevity, let RA = α, Dec. = δ.

$$p(\alpha|\mathrm{d}) = \int_{\delta} \mathrm{d}(\alpha, \delta|\mathrm{d})\, \delta,$$

$$p(\alpha|\mathrm{d}) \pm \sigma(\alpha) = \int_{\delta} (p(\alpha) \pm \sigma(\alpha, \delta))\, \mathrm{d}\delta. \tag{9}$$

In 2D and especially when considering sky localization, we are also interested in the contours that enclose a given volume of probability density to plan optimal observation strategies in the search for electromagnetic counterparts. We propagate the uncertainty estimate produced by the GP (in the space of all realizations from the GP) to the physical parameter space on credible interval contour levels. We define a function, f_{q}, which truncates the posterior density function as follows:

$$f_q(\alpha) = \begin{cases} p(\alpha, \delta|\mathrm{d}) & \text{if } p(\alpha, \delta|\mathrm{d}) \geq q \\ 0 & \text{otherwise} \end{cases}.$$

Such that the integral of f_{q} contains a given proportion of the total probability mass determined by the desired confidence level i.e.

$$\int_{\alpha,\delta} f_q(\alpha, \delta|\mathrm{d})\, \mathrm{d}\delta\, \mathrm{d}\alpha = cl. \tag{10}$$

For a given a confidence level *cl* (usually the 50 and 90 per cent levels), solving equation (10) for q gives q_{cl}, the value of the posterior density of the relevant contour. We obtain the contour, and the error on the contour, by plotting the (RA, Dec.) values for which:

$$p(\alpha, \delta|\mathrm{d}) = q_{cl},$$

$$p(\alpha, \delta|\mathrm{d}) \pm \sigma(\alpha, \delta) = q_{cl}. \tag{11}$$

In the central panel of Fig. 4, we show the samples used to construct the model as well as the 50 and 90 per cent, contours of the GP

[1] See the Data Availability statement 4 for details of the samples we used

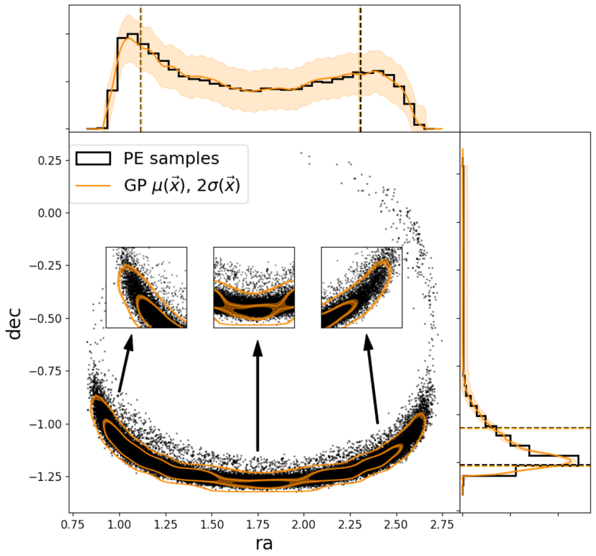

Figure 4. Central panel: Contours of the 2D sky-location of GW150914, the GP model mean prediction and uncertainty (in orange) is compared to the points used to construct the fit (black crosses). Top and left-hand panels show the GP model projections in 1D, compared to the original PE samples. All plots show the 2σ uncertainty around the density estimate as a shaded band.

interpolation in 2D with their respective 2σ uncertainty (the shaded regions). The top and left-hand panels of Fig. 4 show the mean prediction and its 2σ uncertainty marginalized over each parameter by a simple integration of the density over its projection.

The inclusion of the uncertainty highlights several features. On the central inset in Fig. 4, we see that the lower bound on the 50 per cent contour is composed of three islands which correspond to peaks, while for both the mean and the upper bound these islands are connected to obtain a smooth surface at this contour level. For the outer 90 per cent contour, we see that the differences mainly manifest in the tails, where as expected the upper bound follows the well-known *ring* around the sky slightly further. This matches our intuition that there is possibly more density around the ring than around the edges of the contour in the middle of the plot.

4 CONCLUSIONS

We have presented an alternative method for density estimation of marginal PDFs for GW parameters. Our method combines the desirable features of histograms to the extrapolation capabilities of KDEs, within a Bayesian framework. The choice of histogram binning determines the resolution of the PDF, while the kernel of the GP allows the interpolation to be flexible over non-Gaussian correlations and yet smooth. The noise variance parameter of the GP ensures that sources of stochastic noise from the histogram density estimation are taken into account. In cases, where we employ an exact inference scheme, this noise variance can be evaluated for each histogram bin and it is equivalent to heteroskedastic errors over the density estimation. This allows to fully propagate the uncertainty from the PE samples. We plan to extend this method and fully incorporate uncertainties, as we showed in this work for the sky localization example, over higher dimensional posterior surfaces in future work.

This method may be preferable to other methods such as KDEs, a closely related method which is sometimes adopted in the field, depending upon the use-case requirements. It comes with three main advantages: it is suitable for most interpolation problems commonly encountered for GW marginal posteriors; it provides a Bayesian measure of uncertainty over the model predictions; it allows to quickly resample the interpolation using HMC and other samplers available in `TensorFlow`. We presented a series of examples where we know the accuracy of the interpolation is important, such as EOS calculations and sky localization. As the number of events will increase in the next observing run (*O*4), we need reliable tools to post-process the large volume of results.

This work has highlighted the power of GPs to fit a GW posterior surface, a natural extension of this work is to generate a surrogate for the entire likelihood surface, similar to what was done by the authors of Vivanco et al. (2019) using a random forest regressor. Such use of GPs has been already investigated in the field of cosmology to model the Planck18 posterior distribution (McClintock & Rozo 2019). This work has laid the foundation for us to apply a similar methodology to the GW problem in a future work which is currently in preparation. This has applications such as Bayesian quadrature (O'Hagan 1991), efficient jump proposals (Graff et al. 2012; Farr et al. 2020) and more general use of the GP variance to guide the sampling process. The surface learned by the GP can be evaluated directly for a given set of parameters, therefore, avoiding the need to compute expensive waveforms. An example where such likelihood surrogates could be exploited is fast resampling with new astrophysical priors. This could replace an often difficult reweighting procedure, especially when a prior assumption limits the number of available samples in a region of interest (Mandel & Fragos 2020).

ACKNOWLEDGEMENTS

We are grateful to Erik Bodin (Bristol University) and Dr Carl Henrik-Ek (Cambridge University) for useful discussions. We thank Colm Talbot for their useful comments on the manuscript. This work was supported by Science and Technology Facilities Council (STFC) grant ST/V001396/1, and we are grateful for the computational resources provided by Cardiff University and supported by STFC grant ST/V001337/1 (UK LIGO Operations award). This research has made use of data, software, and/or web tools obtained from the Gravitational Wave Open Science Center (https://www.gw-openscience.org), a service of LIGO Laboratory, the LIGO Scientific Collaboration, and the Virgo Collaboration. LIGO is funded by the US National Science Foundation. Virgo is funded by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale Della Fisica Nucleare (INFN), and the Dutch Nikhef, with contributions by Polish and Hungarian institutes. Our code-base is developed upon `TensorFlow` (Abadi et al. 2016) and `GPFlow` (Matthews et al. 2017). Plots were prepared with MATPLOTLIB (Hunter 2007) and the corner plots were made with Corner Foreman-Mackey (2016).

DATA AVAILABILITY STATEMENT

All results in this paper are obtained using publicly available data (Romero-Shaw et al. 2020b) and code (D'Emilio et al. 2021).

REFERENCES

Abadi M. et al., 2016, in 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). p. 265
Abbott B. P. et al., 2016a, Phys. Rev. Lett., 116, 061102
Abbott B. P. et al., 2016b, Phys. Rev. Lett., 116, 241102
Abbott B. P. et al., 2017a, Phys. Rev. Lett., 119, 161101

Abbott B. P. et al., 2018, Phys. Rev. Lett., 121, 161101
Abbott R. et al., 2020a, ApJ, 913, 41
Abbott B. P. et al., 2020b, Living Rev. Relativ., 23, 1
Abbott B. P. et al., 2017b, ApJ, 848, L12
Ashton G. et al., 2019, Astrophys. J. Suppl. Ser., 241, 27
Barrett J. W., Mandel I., Neijssel C. J., Stevenson S., Vigna-Gómez A., 2016, Proc. Int. Astron. Union, 12, 46
Betancourt M., 2017, preprint (arXiv:1701.02434)
Chua A. J., Galley C. R., Vallisneri M., 2019, Phys. Rev. Lett., 122, 211101
Cuoco E. et al. 2020 Mach. Learn.: Sci. Technol.,2, 011002
Cutler C., Flanagan E. E., 1994, Phys. Rev. D., 49, 2658
D'Emilio V. et al., 2021, https://github.com/virginiademi/GP4GW
Del Pozzo W., Berry C. P. L., Ghosh A., Haines T. S., Singer L., Vecchio A., 2018, MNRAS, 479, 601
Fan X., Li J., Li X., Zhong Y., Cao J., 2019, Sci. China Phy. Mech. Astron., 62, 969512
Farr B., Farr W., Rudd D., Price-Whelan A., Macleod D., 2020, Astrophysics Source Code Library. p. ascl–2004
Foreman-Mackey D., 2016, J. Open Source Softw., 1, 24
Gabbard H., Messenger C., Heng I. S., Tonolini F., Murray-Smith R., 2019, preprint (arXiv:1909.06296)
George D., Huerta E. A., 2018, Phys. Lett. B, 778, 64
Graff P., Feroz F., Hobson M. P., Lasenby A., 2012, MNRAS, 421, 169
Green S. R., Simpson C., Gair J., 2020, Phys. Rev. D, 102, 104057
Grover K., Fairhurst S., Farr B., Mandel I., Rodriguez C., Sidery T., Vecchio A., 2014, Phys. Rev. D, 89, 042004
Hensman J., Fusi N., Lawrence N. D., 2013, preprint (arXiv:1309.6835)
Hoy C., Raymond V., 2021, SoftwareX, 15, 100765
Hunter J. D., 2007, IEEE Ann. Hist. Comput., 9, 90
Kanagawa M., Hennig P., Sejdinovic D., Sriperumbudur B. K., 2018, preprint (arXiv:1807.02582)
Khan S., Green R., 2021, Phys. Rev. D, 103, 064015
Landry P., Essick R., 2019, Phys. Rev. D, 99, 084049
Lange J., O'Shaughnessy R., Rizzo M., 2018, preprint (arXiv:1805.10457)
Liu H., Ong Y.-S., Cai J., 2020, IEEE Transactions on Neural Networks and Learning Systems, 32, 708
Lynch R., Vitale S., Essick R., Katsavounidis E., Robinet F., 2017, Phys. Rev. D, 95, 104046
McClintock T., Rozo E., 2019, MNRAS, 489, 4155
McHutchon A., Rasmussen C., 2011, Adv. Neural Inf. Process. Syst., 24, 1341
MacKay D. J. et al., 1998, NATO ASI Series F Computer and System Sciences, 168, 133
Mandel I., Fragos T., 2020, ApJ, 895, 6
Matthews A. G. d. G., Van Der Wilk M., Nickson T., Fujii K., Boukouvalas A., León-Villagrá P., Ghahramani Z., Hensman J., 2017, J. Mach. Learn. Res., 18, 1
Moore C. J., Berry C. P., Chua A. J., Gair J. R., 2016, Phys. Rev. D, 93, 1
Murray I., MacKay D., Adams R. P., 2008, Adv. Neural Inf. Process. Syst., 21, 9
Nair V., Hinton G. E., 2010, Icml
Neal R. M., 2012, Bayesian Learning for Neural Networks. 1, Vol. 118, Springer Science & Business Media,Springer Nature Switzerland
O'Hagan A., 1991, J. Stat. Plan. Inference, 29, 245
Pang P. T., Dietrich T., Tews I., Van Den Broeck C., 2020, Phys. Rev. Res., 2, 033514
Papamakarios G., Pavlakou T., Murray I., 2017, Advances in Neural Information Processing Systems. Inc. Curran Associates, Long Beach, California, USA, p. 2338
Pedregosa F. et al., 2011, J. Mach. Learn. Res., 12, 2825
Pitkin M., Messenger C., Fan X., 2018, Phys. Rev. D, 98, 063001
Quiñonero-Candela J., Rasmussen C. E., 2005, J. Mach. Learn. Res., 6, 1939
Romero-Shaw I. et al., 2020a, MNRAS, 499, 3295
Romero-Shaw I. et al., 2020b, https://dcc.ligo.org/LIGO-P2000193/public
Schmidt P., Ohme F., Hannam M., 2015, Phys. Rev. D., 91, 024043
Setyawati Y. E., Pürrer M., Ohme F., 2020, Class. Quantum Gravity, 37, 075012
Soares-Santos M. et al., 2019, Astrophys. J. Lett., 876, L7
Talbot C., Thrane E., 2020, preprint (arXiv:2012.01317)
Taylor S. R., Gerosa D., 2018, Phys. Rev. D, 98, 1
Thrane E., Talbot C., 2019, Publications of the Astronomical Society of Australia, 36
Veitch J. et al., 2015, Phys. Rev. D, 91, 042003
Virtanen P. et al., 2020, Nat. Method, 17, 261
Vivanco F. H., Smith R., Thrane E., Lasky P. D., Talbot C., Raymond V., 2019, Phys. Rev. D, 100, 103009
Wand M. P., Jones M. C., 1994, Kernel Smoothing. CRC Press, New York
Wang Z., Scott D. W., 2019, Wiley Interdisciplinary Revi. Comput. Stat., 11, e1461
Wang K. A., Pleiss G., Gardner J. R., Tyree S., Weinberger K. Q., Wilson A. G., 2019, Adv. Neural Inf. Process. Syst., 32, 14648
Williams C. K., Rasmussen C. E., 2006, Gaussian Processes for Machine Learning. 1, Vol. 2, MIT Press, Cambridge, MA
Wilson A. G., Hu Z., Salakhutdinov R., Xing E. P., 2016, Artificial Intelligence and Statistics. p. 370

APPENDIX A: TECHNICAL DETAILS OF THE GP MODEL

A1 Data pre-processing

Data pre-processing, often referred to as data set standardization, is a common practice within the realm of machine learning and it can have a very high impact on the accuracy of the model. Our posterior samples have a wide range of values, some having bounds $[-1, 1]$ and some reaching $\mathcal{O}(10^3)$. We rescale our posterior samples such that each parameter ranges between [0, 1] by using the following transformation:

$$\vec{\tilde{\theta}}_\mathrm{d} = \frac{(\vec{\theta}_\mathrm{d} - \min(\vec{\theta}_\mathrm{d}))}{(\max(\vec{\theta}_\mathrm{d}) - \min(\vec{\theta}_\mathrm{d}))}, \tag{A1}$$

where $\widetilde{\theta_d}$ is the vector of transformed samples and the min and max are evaluated for each parameter (i.e. each dimension of the posterior samples vector). The approximate marginalized posterior is scaled according to the *z-score*, such that it has zero mean and unit variance:

$$\widetilde{p}(\theta_i|\mathrm{d}) = \frac{p(\theta_i|\mathrm{d}) - \mu_{p(\theta_i|\mathrm{d})}}{\sigma_{p(\theta_i|\mathrm{d})}}, \tag{A2}$$

where $\widetilde{p}(\theta_i|d)$ is the transformed marginalized posterior, $\mu_{p(\theta_i|d)}$, and $\sigma_{p(\theta_i|d)}$ are the mean and standard deviation of the marginalized posterior points, respectively. All pre-processing, in this work, is performed using `Scikit-Learn` (Pedregosa et al. 2011).

A2 Kernel design

The kernel is defined as the prior covariance between any two function values. Our prior knowledge about the morphology of the posterior can be encoded via this covariance, as it determines the space of functions that the GP sample paths live in. The radial basis function (RBF) or squared exponential kernel is the most basic kernel and it is given as

$$\kappa_{\mathrm{RBF}}(x, x') = \sigma^2 \exp\left(-\frac{1(x - x')^2}{2\ell^2}\right), \tag{A3}$$

where the Euclidian distance between $(x, x^{'})$ is scaled by the length-scale parameter ℓ (measure of deviations between points) and the overall variance is denoted by σ^2 (average distance of the function away from its mean). Functions drawn from a GP with this kernel are infinitely differentiable.

For our application, a more complex kernel architecture that can capture the correlations between parameters is needed. We need smoothness over small-scale features, such that we do not model random noise fluctuations of samples, and flexibility over the large-scale characteristics of the posterior. For this purpose, we employ a combination of RBF and Matern, which is a generalization of the RBF kernel with an additional smoothness parameter ν. The smaller ν, the less smooth the approximated function is

$$\kappa_{\mathrm{M}\nu}(x, x') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{(x - x')}{\ell}\right)^{\nu} K_{\nu} \left(\sqrt{2\nu}\frac{(x - x')}{\ell}\right). \quad \text{(A4)}$$

We choose $\nu = (\frac{1}{2}, \frac{5}{2})$ depending on the specific morphology of the posterior, as this kernel is responsible for encoding its overall shape such as sharp boundary features. The resulting kernel equation is given by

$$\kappa_{GP}(\boldsymbol{\theta}_{\mathrm{d}}, \boldsymbol{\theta}_{\mathrm{d}}{}') = \kappa_{\mathrm{RBF}} \times \kappa_{\mathrm{M52}}.$$

The kernel multiplication corresponds to an element-wise multiplication of their corresponding covariance matrices. This means that the resulting covariance matrix will only have a high value if both covariances have a high value. We also apply automatic relevance determination, which modifies the kernel such that for each dimension an appropriate length-scale is chosen (Neal 2012).

For certain functions, we observe periodicity[2] which can result in a *wrapping* at the period boundary. As mentioned in the paper, this can be encoded into our GP by using a periodic kernel (MacKay et al. 1998). A periodic kernel maps the input dimensions x (e.g. RA in this example) using the transformation $u = [\sin(x), \cos(x)]$ and the original (e.g. the RBF) kernel response is computed in terms of u, this therefore allows one to encode relationships such as wrapping and periodicity. For the standard RBF kernel and a given periodicity, p, the periodic kernel is given by

$$\kappa_{\mathrm{Per(RBF)}}(x, x') = \sigma^2 \exp\left(-\frac{2\sin^2\left(\frac{\pi|x-x'|}{p}\right)}{\ell^2}\right). \quad \text{(A5)}$$

[2] such as the sky location posteriors due to the standard RA, Dec. parameterization

This paper has been typeset from a TEX/LATEX file prepared by the author.