Precessing numerical relativity waveform surrogate model for binary black holes: A Gaussian process regression approach

D. Williams[*] and I. S. Heng
SUPA, University of Glasgow, Glasgow G12 8QQ, United Kingdom

J. Gair
Max Planck Institute for Gravitational Physics,
Potsdam Science Park, Am Mühlenberg 1, D-14476 Potsdam, Germany

J. A. Clark and B. Khamesra
Center for Relativistic Astrophysics and School of Physics,
Georgia Institute of Technology, Atlanta, Georgia 30332, USA

(Received 25 March 2019; revised manuscript received 22 January 2020; accepted 11 February 2020; published 11 March 2020)

Gravitational wave astrophysics relies heavily on the use of matched filtering both to detect signals in noisy data from detectors and to perform parameter estimation and tests of general relativity on those signals. Matched filtering relies upon prior knowledge of the signals expected to be produced by a range of astrophysical systems, such as binary black holes. These waveform signals can be computed using numerical relativity techniques, where the Einstein field equations are solved numerically, and the signal is extracted from the simulation. Numerical relativity simulations are, however, computationally expensive, leading to the need for a surrogate model which can predict waveform signals in regions of the physical parameter space which have not been probed directly by simulation. We present a method for producing such a surrogate using Gaussian process regression which is trained directly on waveforms generated by numerical relativity. This model returns not just a single interpolated value for the waveform at a new point, but a full posterior probability distribution on the predicted value. This model is therefore an ideal component in a Bayesian analysis framework, through which the uncertainty in the interpolation can be taken into account when performing parameter estimation of signals.

DOI: 10.1103/PhysRevD.101.063011

I. INTRODUCTION

The first detection of gravitational waves in September 2015 was the result not only of advanced detector development, but also the development of data analysis techniques which were capable of detecting and characterizing weak signals in noisy data. The most sensitive of these techniques rely on *matched filtering* to identify signals, and these techniques are most effective when accurate and efficient waveform models are available to produce template banks.

The production of high-accuracy waveforms is possible thanks to advances in the field of numerical relativity (NR), in which the full set of Einstein equations are solved numerically. This can be done reliably for the low-mass compact binary systems of interest to the current generation of ground-based gravitational wave observatories; however, these simulations are computationally expensive and can require thousands of CPU hours to run in situations where the mass ratios and spins of the black holes are small. A simulation of a full 350-cycle gravitational waveform spanning the entire advanced LIGO band has been produced [1]; however, this required several months of high-performance computing to complete [2], despite employing numerous techniques to reduce wall-clock computation time. As a result, only around 1000 NR waveforms are available, and most of these are much shorter than 350 cycles long. Binary black hole (BBH) coalescences are described by a number of physical parameters: the ratio of the two component black holes' masses, q; the vector of each component's spin, $\mathbf{s}_1$ and $\mathbf{s}_2$; and the time, t, relative to a fixed reference time, for example, the time of coalescence of the binary.

This results in a parameter space with eight dimensions which is very sparsely sampled. As a result, NR waveforms alone are not a practical way to form the template banks required for precise signal parameter estimation. In addition, the high cost of producing new simulations is unlikely to significantly change this situation in the near future.

To overcome this problem, there have been significant efforts to inform analytical models of nonspinning black hole coalescences with the results of NR simulations of spinning systems to produce an analytical, phenomenological approximant which can be rapidly evaluated. There are two major implementations of analytical models which

[*] daniel.williams@glasgow.ac.uk

are calibrated against NR-derived waveforms, the Phenom and SEOBNR families of approximants. The Phenom family has developed from IMRPhenomA [3], which was capable of producing waveforms for nonspinning binaries, through to IMRPhenomD [4], which models spinning, nonprecessing binaries.

The Phenom family of waveforms has been developed to incorporate support for precessing systems through the IMRPhenomP codes; the latest edition of this is IMRPhenomPv3 [5], although in this work we will make use of the slightly older IMRPhenomPv2 [4], which has extensive support within the PyCBC [6–8] library used in the preparation of this work. This is composed of a post-Newtonian approximation to the inspiral period of the waveform and a phenomenological ansatz for the merger and ringdown periods. The approximant is calibrated against 19 NR-derived waveforms to produce a model which has a low mismatch [defined in Eq. (8)] with the calibration data.

The SEOBNR family provides an alternative approach to that taken by the IMRPhenomP models, using an effective one-body approach [9–11] to map the dynamics of a binary into those of a single test particle in a deformed Kerr metric. In contrast to the piece-wise approach to building the waveform from the inspiral, merger, and ringdown of the IMRPhenomP models, the SEOBNR models construct the waveform through a single process [12]. A number of models based on the effective one-body approach exist, ranging from EOBNR which model nonspinning systems [12,13] to the SEOBNR families of model, which can model spinning systems [14–16], and precession effects [17]. Similarly to IMRPhenom, these models are calibrated against NR waveforms: for SEOBNRv3, five waveforms are used for this calibration.

These models can be evaluated quickly and are thus suitable for the rapid parameter estimation tasks required for the detection and characterization of gravitational waves. However, both the Phenom and SEOBNR models are affected by systematic uncertainties which are difficult to quantify in regions of the BBH parameter space which are not calibrated against NR simulations.

The NRSur family of surrogate models, developed by Blackman *et al.* [18–20], employs spline interpolation to waveforms generated by the SpEC NR code. The two analysis-ready versions of this model, NRSur4d2s and NRSur7d2s, are capable of producing waveforms for systems with a mass ratio < 2 and an effective spin parameter < 0.8. In contrast to phenomenological models, the NRSur models are currently capable of producing only a small number of cycles of the waveform, being limited by the length of the NR waveforms off which they are conditioned. Recent efforts have been made, however, to produce similar surrogate models which are conditioned on hybridized waveforms [21]. The number of waveforms required to produce the surrogate model is also considerably larger than those required for the phenomenological models, with NRSur7d2s being conditioned on 744 NR waveforms.

Efforts to account for the systematic uncertainty between NR waveforms and waveforms produced by phenomenological models have been proposed in which the uncertainties are modeled by Gaussian process regression (GPR) [22,23]. This allows the uncertainty in the interpolation to be calculated from the posterior predictive distribution of the GPR. This probability distribution, derived from GPR, can be used to marginalize the likelihood of the observed gravitational wave (GW) data over waveform uncertainty. This approach was shown to provide a significant reduction in biases in parameter estimation (PE) compared to using phenomenological methods with no attempt to account for the uncertainty [22,23].

These previous efforts suggested using GPR to model the difference between NR waveforms and phenomenological models. We propose to extend this approach by producing a model of the entire gravitational waveform using GPR as a surrogate model conditioned only on numerical relativity simulations, without any reference to a phenomenological model. In comparison to the NRSur families of surrogate, GPR is capable of not only producing an approximant for the waveform throughout the parameter space, but also an uncertainty on that estimate. We note that our model is not the first to attempt to predict BBH waveforms using GPR, but it is the most complete. A previous model [24] used GPR, but this was conditioned on waveforms generated from the IMRPhenomPv2 phenomenological approximant, and not NR data, and is not capable of producing generically spinning waveforms.

GPR is a Bayesian regression technique which relies on a Gaussian process (GP) prior distribution. A GP can be considered as a prior over a space of functions, each of which is considered a potential fitting function to some set of data. The GP model assumes that the values of the function evaluated at a certain finite set of points are draws from a multivariate Gaussian distribution. The GP prior is itself defined by a number of assumptions about the behavior of these functions (e.g., their smoothness). When the GP prior model is conditioned on data from existing simulations (potentially allowing for uncertainties in each of the simulations), the resulting posterior provides a distribution of functions which could represent the true model. The mean of this posterior distribution can be used analogously to the single fitting function which is produced by more conventional regression techniques, while the variance of the distribution provides a measure of the goodness-of-fit.

The structure of this publication is as follows. In Sec. II, we explain the process used for the production of a waveform surrogate model and the choice of covariance function for our model in Sec. II A. Our new model, named Heron, is introduced in Sec. III, with the waveforms used to train the model described in Sec. III A, and a discussion of the complications introduced by using a large quantity of data is provided in Sec. III B. An overview of the testing procedures which we used to verify the output of the model is included in Sec. IV, with both the results of these tests, and a number of example waveform outputs are presented in Sec. V.

II. GAUSSIAN PROCESS REGRESSION

A GP represents a distribution of potential functions which can explain a set of training data $(\mathcal{X}, \mathcal{Y})$, composed of observations, $\mathcal{Y}$, made at locations, $\mathcal{X}$, within the parameter space of the problem, such that the function values

$$y = f(\boldsymbol{x})$$

(for each $\boldsymbol{x} \in \mathcal{X}$, $y \in \mathcal{Y}$), are modeled as being drawn from a multivariate normal distribution. As such, the GP is fully characterized by its mean function, $\mu(\boldsymbol{x})$, and a covariance function, $k(x, x')$, which describes the similarity between two function values at two points in the parameter space. A GP can be defined with any positive-definite covariance function, the form of which encodes prior assumptions about the data, for example, its smoothness and stationarity. Popular choices of covariance function include the squared exponential covariance functions and Matérn covariance functions [23,25].

It is common to assume the training samples have mean zero. This causes the mean of the GP to be zero outside the training set, which, while unphysical, is a reasonable assumption given a lack of data; within the region described by the training set, the mean of the function is defined by the training data. Making this assumption allows the mean squared properties of the data to be determined entirely through the covariance function.

When defining the covariance function for the GP, it is often desirable to specify a number of free hyperparameters, $\boldsymbol{\theta}$, which allow the properties of the covariance function (and hence the GP) to be adapted based on the training data. Bayesian model comparison can be used to select the GP which optimally describes the data, or to obtain a posterior distribution on appropriate values of the hyperparameters. The log-probability that a given set of function values were drawn from a GP with zero mean and a covariance matrix $K_{ij} = k(x, x')$ is

$$\log(p(\boldsymbol{y}|X)) = -\frac{1}{2}\boldsymbol{y}^{\mathsf{T}} K^{-1} \boldsymbol{y} - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi. \quad (1)$$

With n, the total number of points in the training data. This quantity is normally denoted as the *log-evidence* or the *log-hyperlikelihood*. The model which best describes the training data may then be found by maximizing the log-hyperlikelihood with respect to the hyperparameters, θ of the covariance function.

Once the GP has been conditioned on the training data and the optimal covariance function identified through model comparison, it is possible to exploit it as a predictive tool, allowing the interpolation of function outputs between training data. In order to make a prediction using the GP model, we require a new input point at which the prediction should be made, which is denoted x^*. In order to form the predictive distribution, we must then calculate the covariance of the new input with the existing training data, which we denote K_{x,x^*}, and the autocovariance of the input, K_{x^*,x^*}. We then define a new covariance matrix, K^+, which has the block structure

$$K^+ = \begin{bmatrix} K_{x,x} & K_{x,x^*} \\ K_{x^*,x} & K_{x^*,x^*} \end{bmatrix} \quad (2)$$

for $K_{x,x}$ the covariance matrix of the training inputs, and $K_{x^*,x} = K_{x,x^*}^T$.

The predictive distribution can then be found as

$$p(\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D}) = \mathcal{N}(\boldsymbol{y}^*|K_{x^*,x}K_{x,x}^{-1}\boldsymbol{y}, K_{x^*,x^*} - K_{x^*,x}K_{x,x}^{-1}K_{x,x^*}), \quad (3)$$

where $\mathcal{D}$ is the training data and $\mathcal{N}$ is the normal distribution.

Equation (3) emphasizes the value of the GP approach to interpolation, as the value returned from the model is not a single point prediction, but a posterior probability distribution which describes the uncertainty of the prediction, along with the "best estimate" prediction as the mean of $p(\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D})$.

A. Choice of covariance function

A covariance function can be designed for any given GP by considering both the hyperparameters and functional form of the covariance function. A much fuller discussion of these considerations is given in [23]; however, a summary is made here due to the importance of these considerations in the remainder of this work.

A covariance function must be positive definite, that is, it returns a value which is non-negative for any element in its domain. Practically, when working with data, this means that the covariance function will map any pair of points in the set of data to a non-negative real number. We can additionally require a covariance function to be stationary, in which case it is a function of $x_1 - x_2$, and so invariant to translations in the input space. Further, if it is a function of $|x_1 - x_2|$ only it is an isotropic covariance function, and invariant under rigid motions within the input space [25].

A straightforward function of $x_1 - x_2$ is a distance function of the form

$$d^2(x_1, x_2) = \sum_{a,b} (x_1 - x_2)^a (x_1 - x_2)^b. \tag{4}$$

Such a distance function is stationary, and a covariance function using this distance metric will then be a stationary GP.

The functional form of the covariance function is important in defining the prior belief about the form of the function which generated the training data. A common choice of covariance function is the exponential squared covariance function [25],

$$k_{\mathsf{SE}}(d; \lambda) = \exp\left(-\frac{d^2}{2\lambda^2}\right). \tag{5}$$

For λ, the length scale of the kernel can be tuned as a hyperparameter. A larger value of this parameter will describe longer scale variations within the data.

The functional form of the squared exponential covariance function implies that the generating function was infinitely differentiable; however, generalizations of this covariance function allow the differentiability to be altered through the addition of a further hyperparameter, allowing the smoothness of the generating function to be learned during the training of the GP.

An example of such a covariance function is the general Matérn covariance function, which has the form

$$C_\nu(d; \rho, \nu) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{d}{\rho}\right)^\nu K_\nu\left(\sqrt{2\nu}\frac{d}{\rho}\right) \tag{6}$$

for Γ the gamma function, K_ν the modified Bessel function of the second kind, and ρ and ν are hyperparameters. A GP which uses this covariance function will be $(\nu - 1)$-times differentiable [25].

Uncertainty in the training data used to train the GP can be accounted for by modifying the covariance matrix appropriately, with K^+ of Eq. (2) becoming

$$K^+ = \begin{bmatrix} K_{x,x} + \sigma_i^2 I & K_{x,x^*} \\ K_{x^*,x} & K_{x^*,x^*} \end{bmatrix}, \tag{7}$$

for I the identity matrix, and σ_i the standard deviation of the ith datum.

The predictive distribution then becomes

$$p(\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D}) = \mathcal{N}(\boldsymbol{y}^*|K_{x^*,x}(K_{x,x} + \sigma_i^2 I)^{-1}\boldsymbol{y}, \\ K_{x^*,x^*} - K_{x^*,x}(K_{x,x} + I\sigma_i)^{-1}K_{x,x^*}).$$

The inclusion of a small noise term, by setting σ to a small value, such as 10^{-6}, is often advantageous for improving the numerical stability of the inversion of the covariance matrix (Tikhonov regularization), which can otherwise become nearly singular as the total amount of training data increases.

More complex covariance models can be obtained by combining simpler covariance functions through addition or multiplication. This allows the modeling of effects within the training data which occur at different scale lengths or with different properties. For example, if the training data are produced by a process with a long-term variation, but within that long-term variation there are a number of short-term variations, we might model this as a combination of two covariance functions, specifically the sum of two exponential squared covariance functions. Similarly, it is possible to define a GP that uses different kernels in different dimensions of the parameter space, allowing the scale length of each dimension to be chosen individually; for this purpose, we might use a kernel that is a product of different kernels for each dimension. In the case of a diagonal metric, this happens automatically when using the squared-exponential covariance function, and covariance functions with similar form, since they determine the scale of each dimension independently.

B. Training the surrogate

Then, in order to produce a good fit to the data, and to accurately estimate the uncertainty of the prediction from the regression model, we performed Bayesian model selection to determine the optimal value of the covariance function's hyperparameters. In order to initialize this process, we made a rough guess of appropriate values for the hyperparameters; we do this by calculating the average distance between points along each axis in the data space and using this as our initial estimate for the hyperparameter values. Starting from these initial values we optimized the log-likelihood of the model by varying the hyperparameter values to determine a maximum *a posteriori* log-likelihood.

In order to cope with the large number of training points and to increase the speed of the training process, we used the ADAM [26] optimization algorithm to stochastically optimize the log-likelihood using minibatches of 100 training points.

While this method of determining, and fixing, the hyperparameters of the GP is computationally convenient, other methods are also possible, including marginalizing over the hyperparameters. However, our method has the advantage that it is not necessary to evaluate the GPR model for all of the hyperparameter samples and can therefore be evaluated more rapidly.

III. THE HERON MODEL

Using a GPR model, named Heron, trained on NR data from the Georgia Tech BBH waveform catalog. Heron was designed as a surrogate model operating over the eight dimensions of the BBH parameter space, and we present it

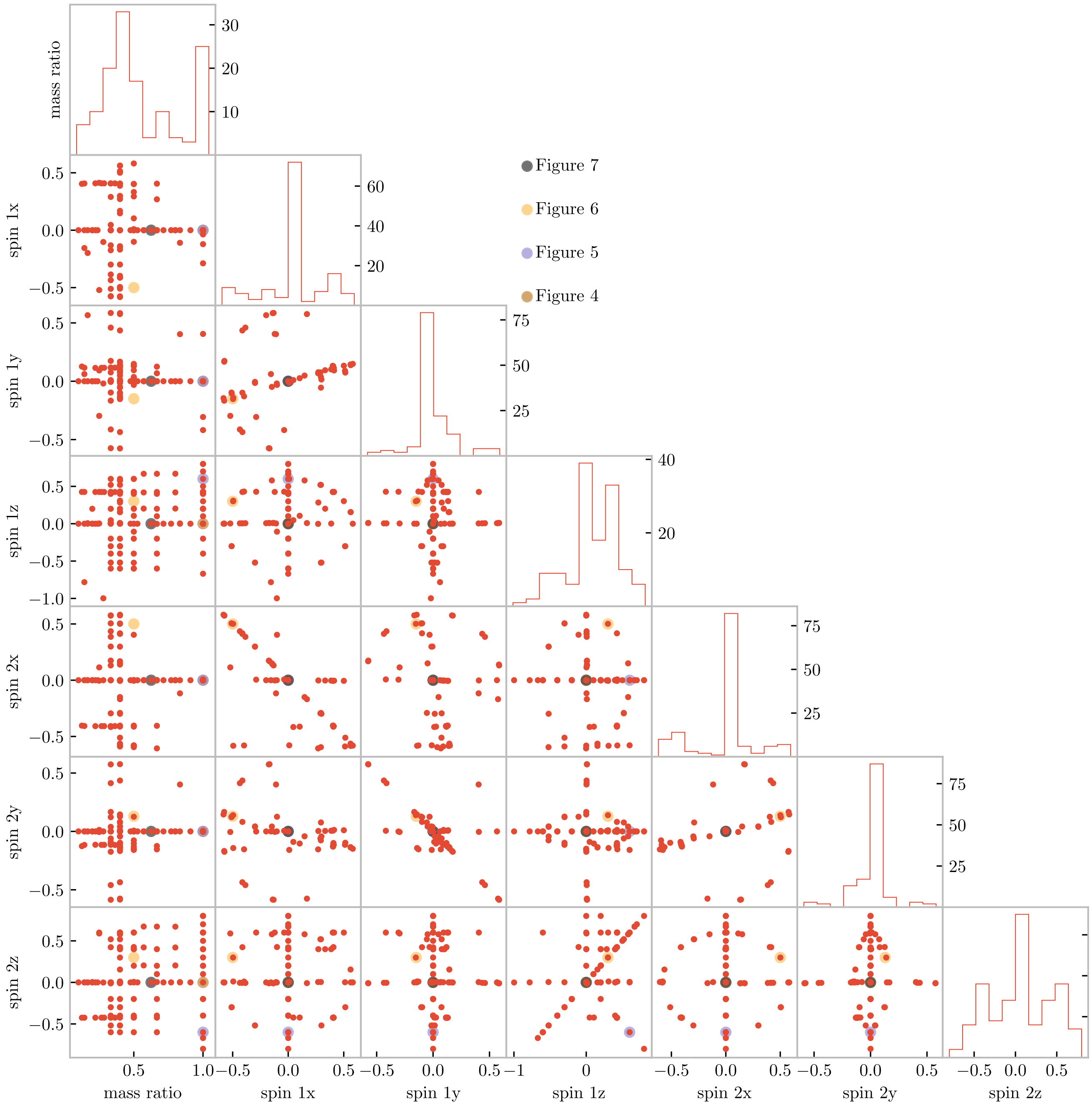

FIG. 1. A pair plot of the parameter space sampling in the Georgia Tech catalog. The subplots on the diagonals show the histograms of the distribution of waveforms (as red points) generated with respect to each individual parameter. Three additional points are displayed on the plot corresponding to the waveform samples shown in the later figures of this paper.

as a proof of concept of a GPR-based surrogate for this larger parameter space. The model is constructed using a squared-exponential covariance function. We will demonstrate that this model is capable of producing waveforms for spinning and precessing BBH systems.

A. Training data

We constructed our training data for the Heron model from the strain values of the 132 waveforms in the Georgia Tech Catalog [27]. These data were acquired in the LIGO numerical relativity hdf5 format [28], and the PyCBC package [6–8] was used to produce the (2,2)-mode of these waveforms.

Each waveform is parametrized by seven quantities (the mass ratio and the spin vectors of each component black hole) in a vector we denote $\boldsymbol{x}_i$. Each strain value, h_i, within the waveform is further parametrized by a time relative to the maximum strain value in the waveform, and thus each training point is parametrized by an eight-dimensional parameter vector, which we denote $\boldsymbol{x}_i{}'$. This provides us with a training set which has eight input dimensions, and a single output dimension, with the form

$$\mathcal{D} = \{(\boldsymbol{x}_i{}', h_i) | i = 1, 2, \ldots, N\}$$

for N the total number of strain samples used from all of the training waveforms. The distribution of training waveforms throughout the parameter space is shown in Fig. 1.

B. Computational complexity

A major drawback of GPR is the need to invert the covariance matrix in order to produce predictions. Matrix inversion is a computationally intensive task which scales in memory with N^2, for N training points, and with N^3 in time. The standard approach to GPR described in Eq. (3) thus rapidly becomes impractical, requiring large quantities of memory for even moderately sized training sets. In order to overcome these scaling problems, approximate GPs simplify the inversion of the covariance matrix by making simplifying assumptions about its form. One example is the use of the approximate hierarchical off-diagonal low rank (HODLR) [29] inversion method, which allows inversion to be carried out in $\mathcal{O}(N \log^2 N)$ operations. This approach is possible because kernels such as the exponential squared kernel produce covariance matrices which can be arranged to form HODLR matrices. The off-diagonal blocks are then factorized using partial-pivoted lower-upper decomposition, and the on-diagonal blocks are factorized using a more accurate algorithm, such as Cholesky decomposition. The block inverses are then recombined to provide the (approximate) overall matrix inverse.

This surrogate model makes use of $N = 4,740$ training points, stored as 64-bit floating points, and requires approximately 370 kilobytes to store in memory. This leads to the need to invert a covariance matrix which requires around 134 gigabytes of memory. To overcome this, we employed the HODLR method for calculating the matrix inverse, using the implementation in the GEORGE [29] PYTHON package.

The use of an approximate method to produce the GP posterior will introduce additional uncertainties. While tests conducted in [29] indicate that this additional uncertainty is likely to be small, we make use of in-sample testing (see Sec. IVA) to assess the impact of using this method on the model's ability to replicate its training data.

IV. VERIFICATION OF THE GPR MODEL

The sparsity of training data poses a considerable challenge to the testing and verification of a model such as the Heron model; conventional approaches to testing such a model involve setting aside a fraction of the training data to compare to the model output when evaluated at the parameter space location of each test datum.

The quantity of numerical relativity waveforms available at present in the Georgia Tech catalog makes this approach difficult, as some regions of the parameter space are very sparsely sampled, and omitting a training waveform in this location may significantly complicate the process of training the model. To overcome this, we have carried out three separate categories of test on the Heron model.

In-sample tests where the entire catalog of available training waveforms are used to condition the GPR used by the model. Waveforms are then produced from the model at the parameter locations which correspond to each of the training waveforms, and the match between the Heron waveform and the GPR waveform is calculated.

Out-of-sample tests where a single waveform from the catalog is omitted from the set of training waveforms used to condition the GP. A GPR model is conditioned on a reduced catalog for each waveform, the model is retrained to find the optimal hyperparameters given the reduced dataset, and the waveform is produced from the reduced Heron model which corresponds to the omitted NR waveform. The match is then computed between these two waveforms.

Tests against phenomenological models where the match is computed between waveforms produced by Heron and by other waveform models, such as SEOBNRv3 and IMRPhenomPv2.

Each approach to testing has different advantages and disadvantages and test for different aspects of the model's performance.

For each of the tests, we compare the output of the Heron model with another waveform by calculating the mismatch between the two waveforms. This is defined as

$$\mathcal{M}(h_{\text{model}}, h_{\text{ana}}) = 1 - \max_{t_0, \phi_0} \frac{\langle h_{\text{model}}, h_{\text{ana}} \rangle}{\sqrt{\langle h_{\text{model}}, h_{\text{model}} \rangle \langle h_{\text{ana}}, h_{\text{ana}} \rangle}}, \tag{8}$$

where h_{model} and h_{ana} are, respectively, the timeseries predicted by the GPR model and the analytical phenomenological approximant, t_0 and ϕ_0 are the merger time and merger phase, and $\langle \cdot, \cdot \rangle$ is the noise-weighted inner product between two waveforms, defined as

$$\langle a, b \rangle = \Re \int_{-\infty}^{\infty} \frac{\tilde{a}^*(f) \tilde{b}(f)}{S_n(f)} \mathrm{d}f \tag{9}$$

for $\tilde{a}$ and $\tilde{b}$, respectively, the Fourier transforms of the timeseries a and b, S_n the amplitude spectral density of the noise, and f the frequency.

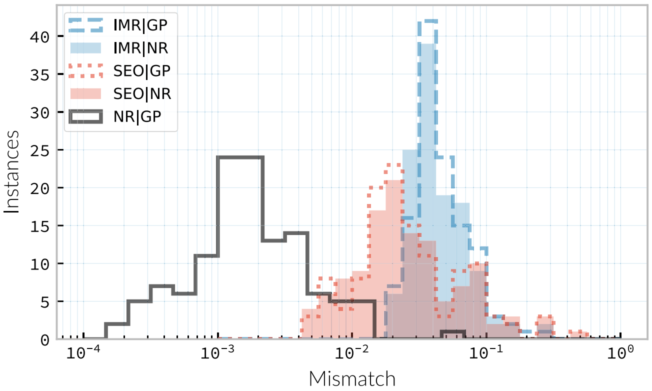

FIG. 2. The distributions of mismatches between waveforms from the Heron model and each of the NR waveforms from the Georgia Tech waveform catalog (black outline histogram) used in the training set using the procedure described in Sec. IV A. Additionally, the mismatch distributions between waveforms produced at the same parameters as the NR waveforms by the SEOBNRv3 (red outline histogram) and the IMRPhenomPv2 (blue outline histogram) phenomenological waveform models are plotted. For comparison, the distributions of mismatch between the same Georgia Tech waveforms and the corresponding waveforms from the SEOBNRv3 and IMRPhenomPv2 models are plotted as filled red and blue histograms, respectively.

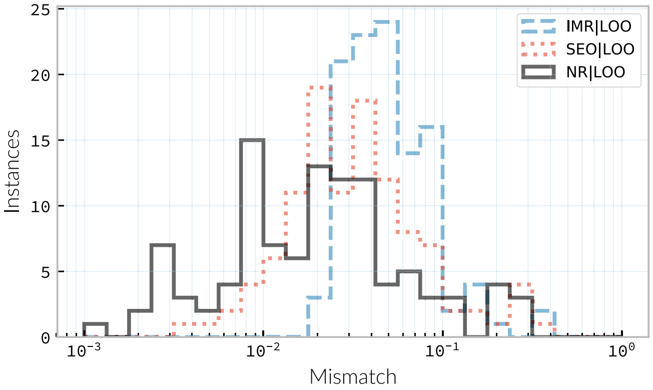

FIG. 3. The distributions of mismatches between waveforms from the Heron model and each of the NR waveforms from the Georgia Tech waveform catalog (black outline histogram) used in the training set using the LOO testing procedure detailed in Sec. IV B. Additionally, the mismatch distributions between waveforms from the Heron model and waveforms produced at the same parameters as the NR waveforms by the SEOBNRv3 (red outline histogram) and the IMRPhenomPv2 (blue outline histogram) phenomenological waveform models are plotted.

For all of the tests presented in this paper, we assume that the noise is flat across frequencies, that is $S_n(f) = 1 \ \forall \ f$.

A. In-sample tests

The simplest set of tests which we perform on the Heron model are *in-sample* tests, which effectively test the model's ability to reproduce its own training data. For the Heron model, this involved computing the mean waveform from the GP corresponding to each waveform which was used in the training set. The mismatch was then calculated between each mean waveform and the corresponding NR training waveform using the expression for waveform mismatch, $\mathcal{M}$, given in Eq. (8).

In-sample testing ought to reveal problems with the choice of hyperparameters in the model, inconsistencies in the training data itself, and error introduced into the model through the use of an approximate method for the inversion of the covariance matrix. Figure 2 plots the histogram of the mismatch (equal to $1 - \mathcal{M}$) values which resulted from these tests against the Georgia Tech waveforms used as the training data (plotted as the black-outlined histogram). Reassuringly, the mismatch between the vast majority of the model outputs and the training data is small. The mean mismatch from these in-sample tests is 0.003, with 95% of the mismatches falling between 0.000245 and 0.0124. This implies that the additional error introduced into the waveform using the approximate matrix inversion technique is responsible for only a small mismatch when compared to the NR waveform.

B. Out-of-sample tests

A more rigorous test of a predictive model involves comparing the model's output in a region of the parameter space which does not contain a training datum. This process, known as out-of-sample testing, is difficult for the Heron model, thanks to the large (seven dimensional) parameter space, and the small number of available training waveforms. As a result, removing a substantial fraction of the waveforms in order to produce a set of test data would be likely to substantially affect the predictive power of the model.

To overcome this, we performed a leave-one-out (LOO) testing procedure. In order to do this, multiple training datasets are produced; from each, a single waveform is omitted. This reduced dataset is then substituted for the data on which the full Heron model's GP is conditioned, and the model is retrained using the reduced training set, in order to find the hyperparameter values which maximize the model's log-likelihood. The reduced Heron model is then evaluated at the parameter location corresponding to the omitted waveform, in order to compute a predicted mean waveform. The mismatch between the predicted waveform and the omitted NR waveform was then computed, and the distribution of these mismatches is plotted in Fig. 3 as a black-outlined histogram.

The mean mismatch across all of the tests was 0.0369, with 95% of the mismatches between 0.000922 and 0.226. A total of eight tests produce a mismatch greater than 0.1, and in each case the variance of the returned waveform is very large, indicating that the model is able to express its lack of knowledge about these regions of the parameter space effectively. While this uncertainty

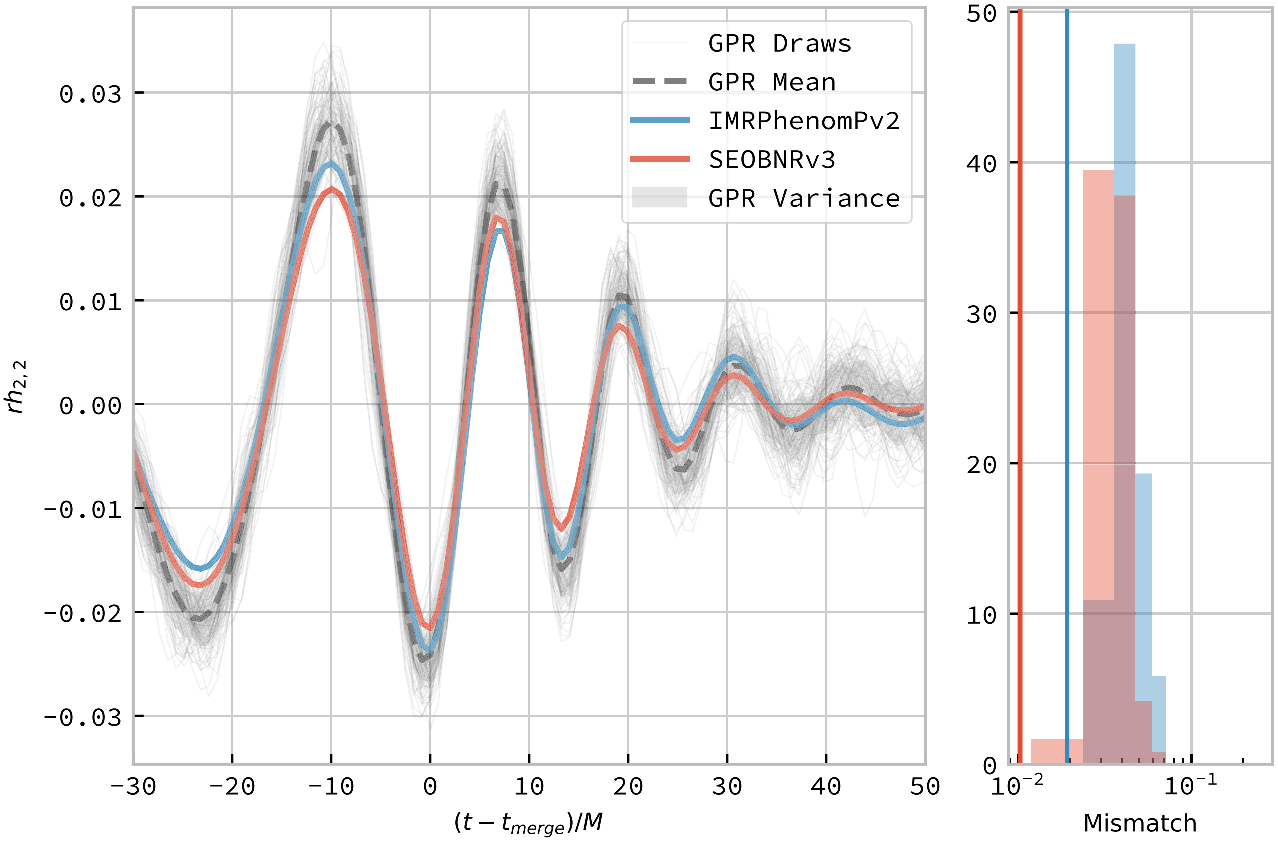

FIG. 4. Nonspinning waveform. One hundred draws from the Gaussian process (left panel) for a nonspinning, equal-mass configuration ($\boldsymbol{s}_1 = (0,0,0)$, $\boldsymbol{s}_2 = (0,0,0)$, $\boldsymbol{q} = 1.0$), shown as light gray lines compared to two analytical phenomenological approximant models, SEOBNRv3 and IMRPhenomPv2 in red and blue, respectively. The mean draw from the Gaussian process is shown as a gray dashed line, while the associated variance is plotted as a gray-filled region surrounding the mean. In the right panel, the distribution of mismatches between the samples and both phenomenological waveforms is shown, with the vertical lines representing the mismatch between the mean waveform from the GPR and the phenomenological waveform.

could be directly incorporated into some applications of the model, it could also be used to automatically flag draws from the model which are of low confidence and which should not be relied upon in an analysis.

C. Tests against phenomenological models

It may also be helpful to understand how the outputs of the Heron model compare to conventional phenomenological approximants which are in widespread use. To do this, we calculated the mismatch between the output of the Heron model at the same parameter locations as the in-sample and leave-one-out tests.

In the left panel of Figs. 4 and 5, we compare the waveform computed for different random samples drawn from the GPR model, the mean of the GPR model, and the IMRPhenomPv2 and SEOBNRv3 waveforms for a nonspinning configuration (Fig. 4), an equal-mass configuration with antialigned spins (Fig. 5), and a precessing configuration (Fig. 6). The distribution of mismatches between the GPR model predictions and the two phenomenological approximants is shown in the right panel of each figure, with matches calculated between the approximant waveforms and 100 sample waveforms drawn from the GPR model. In addition, the mismatch between the mean waveform produced by the GPR model and each phenomenological model is indicated by a solid line; it is noteworthy that this mismatch is always smaller than the mean of the mismatches between the sample draws and the phenomenological models. This is a result of the mismatch being a somewhat asymmetric indicator: the mismatch will always be higher for a waveform which overestimates or underestimates some feature of the waveform, where the over- and underestimates will be averaged through the use of the mean waveform, producing a lower mismatch.

In the in-sample case, the Heron model reproduces the NR waveforms with substantially lower mismatch than either phenomenological model. This behavior is to be expected, since the Heron model has direct access to the NR data, where the phenomenological models do not. It is worth noting that the mismatch for SEOBNRv3 is consistently smaller than that of IMRPhenomPv2 against both NR

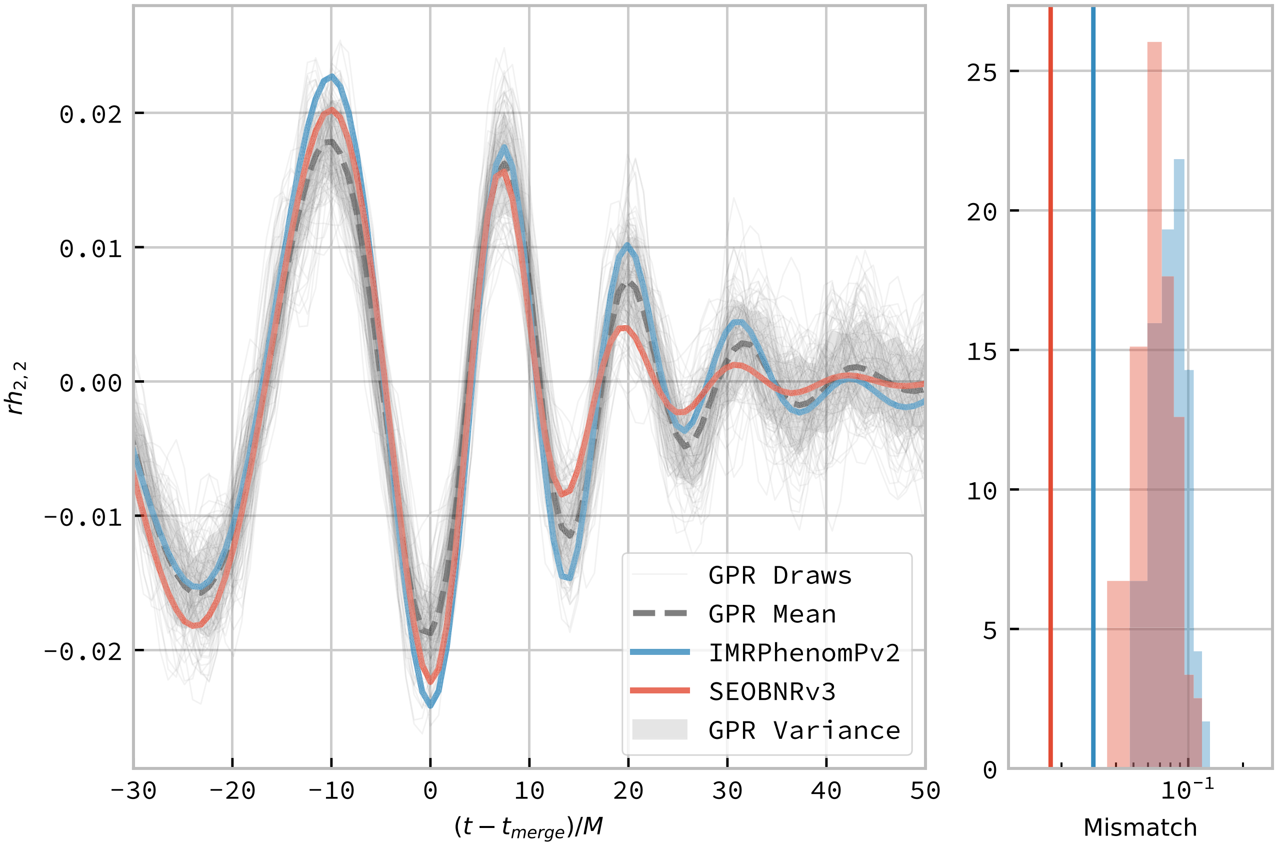

FIG. 5. Antialigned spin waveform. One hundred draws from the Gaussian process (left panel) for a nonspinning, equal-mass configuration ($\boldsymbol{s}_1 = (0, 0, 0.6)$, $\boldsymbol{s}_2 = (0, 0, -0.6)$, $\boldsymbol{q} = 1.0$) shown as light gray lines compared to two phenomenological approximant models, SEOBNRv3 and IMRPhenomPv2 in red and blue, respectively. The mean draw from the Gaussian process is shown as a gray dashed line, while the associated variance is plotted as a gray-filled region surrounding the mean. In the right panel, the distribution of mismatches between the samples and both phenomenological waveforms is shown, with the vertical lines representing the mismatch between the mean waveform from the GPR and the phenomenological waveform.

and the Heron model. IMRPhenomPv2 is known to be accurate over a smaller range of black hole spins than the SEOBNRv3 model.

We also compare the behavior of the LOO models described in Sec. IV B with the two phenomenological models. The distributions of mismatches from comparison between waveforms from the LOO models and waveforms produced by each approximant at the same parameter location as the NR waveform which was omitted from the LOO model are plotted in Fig. 3 as blue and red-outline histograms for the IMRPhenomPv2 and SEOBNRv3 waveforms, respectively. Here we see that the LOO models are generally in good agreement with the two approximants, with the mismatches slightly larger between the LOO models and the approximants than between the LOO models and the NR waveforms, which is also seen in the in-sample testing.

V. EXAMPLE WAVEFORMS

While we have discussed at length the various tests which we carried out on the Heron model, it is valuable to be able to visually compare the output of this model with the phenomenological models used in testing.

In the left panel of Figs. 4 and 5, we compare the waveform computed for different random samples from the GPR model, the mean of the GPR model, and the IMRPhenomPv2 and SEOBNRv3 waveforms for a nonspinning configuration (Fig. 4), and an equal-mass configuration with antialigned spins (Fig. 5). The distribution of mismatches between the GPR model predictions and the two phenomenological approximants is shown in the right panel of each figure, with matches calculated between the approximant waveforms and 100 sample waveforms drawn from the GPR model.

An example of a precessing waveform generated by the GPR model is also shown in Fig. 6.

In Fig. 7, we also show one of the training NR waveforms plotted alongside the mean output of the GPR model, 100 waveform draws from the model, and waveforms produced from both of the phenomenological models used for the comparisons in Figs. 4–6.

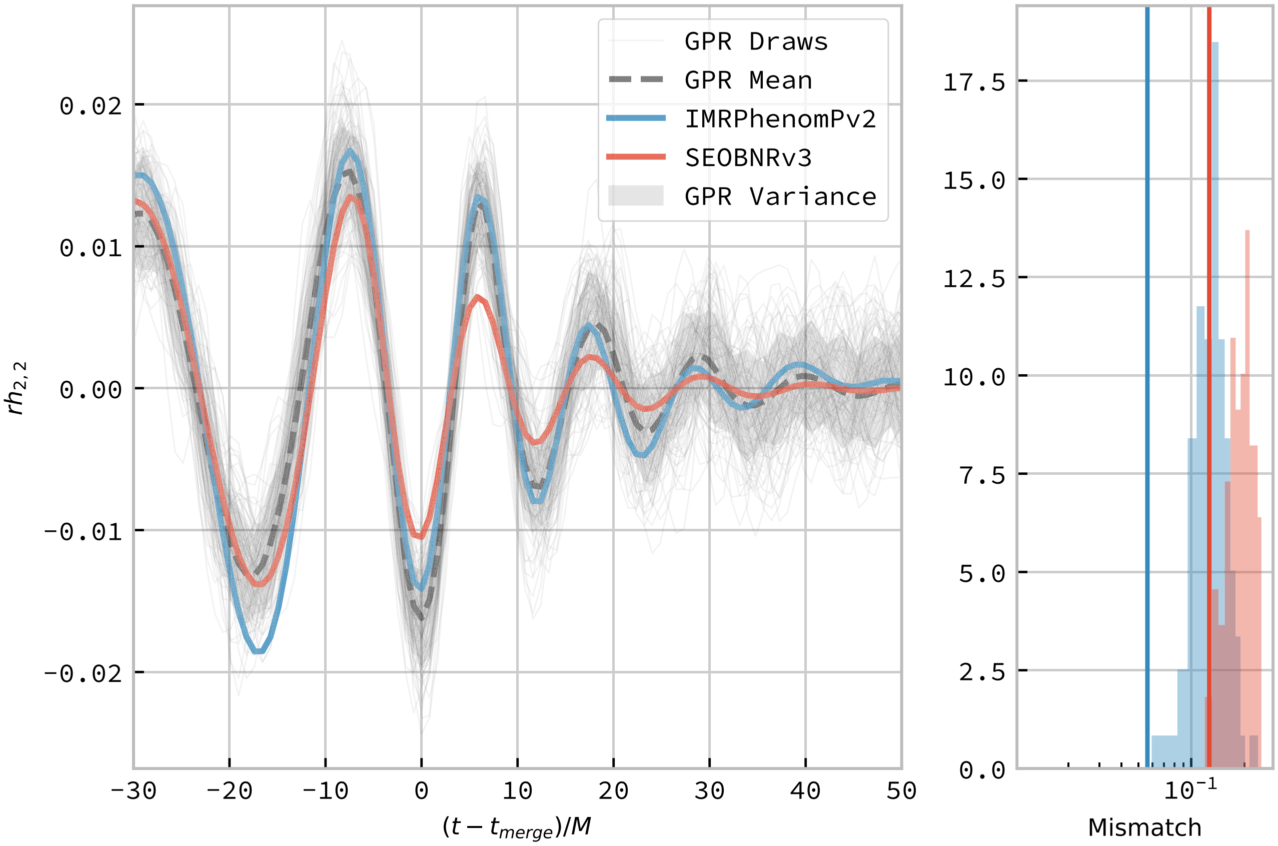

FIG. 6. Precessing waveform. One hundred draws from the Gaussian process (left panel) for a precessing system, with a mass ratio $q = 0.4$, and a spin configuration ($\boldsymbol{s}_1 = (-0.5, -0.15, 0.3)$, $\boldsymbol{s}_2 = (0.5, 0.13, 0.3)$), shown as light gray lines compared to two phenomenological approximant models, SEOBNRv3 and IMRPhenomPv2 in red and blue, respectively. The mean draw from the Gaussian process is shown as a gray dashed line, while the associated variance is plotted as a gray-filled region surrounding the mean. In the right panel, the distribution of mismatches between the samples and both phenomenological waveforms is shown, with the vertical line representing the mismatch between the mean waveform from the GPR and the phenomenological waveform.

VI. SUMMARY

We have entered the era of routine GW detection, and the ability to accurately and rapidly characterize signals from events such as BBH coalescences will be critical to understanding the properties of these systems. This characterization process relies on the availability of waveform templates which are either precomputed prior to the analysis being run, or can be generated on-the-fly. Highly accurate waveforms, generated by NR simulations, are able, and in principal can facilitate accurate inference on detected signals. However, the expense of producing them limits their coverage of the parameter space; as a result of this lack of coverage, and the considerable time requirements to produce new waveforms, any inference method which relied solely on NR techniques could not hope to satisfy the requirement to rapidly characterize signals and would not be practical in a scenario where multiple events are detected every month. Phenomenological models, which can be evaluated rapidly, are available, which attempt to interpolate across a large volume of the parameter space, but the accuracy of the waveforms which they produce can be difficult to assess. This leads to the possibility of introducing biases into the inferred properties of the system which generated the signal.

In this paper, we have laid out an approach to improving the accuracy of gravitational wave parameter estimation in the context of limited template availability by implementing a waveform approximant model using GPR, providing not only a point estimate of the waveform at any point in the BBH parameter space, but also a distribution of plausible waveforms, allowing the uncertainty of the interpolation to be taken into account during the analysis. In contrast to previous attempts to produce a GPR model for GW waveforms, such as [24], our model is trained on data from the Georgia Tech NR waveform catalog, described in Sec. III A.

We introduced GPR in Sec. II as a nonparametric regression method. This property allows the regression model to be constructed while making minimal assumptions about the form of the waveforms, which are encoded through the form of the covariance function. We discuss covariance functions in Sec. II A. In order to reduce the

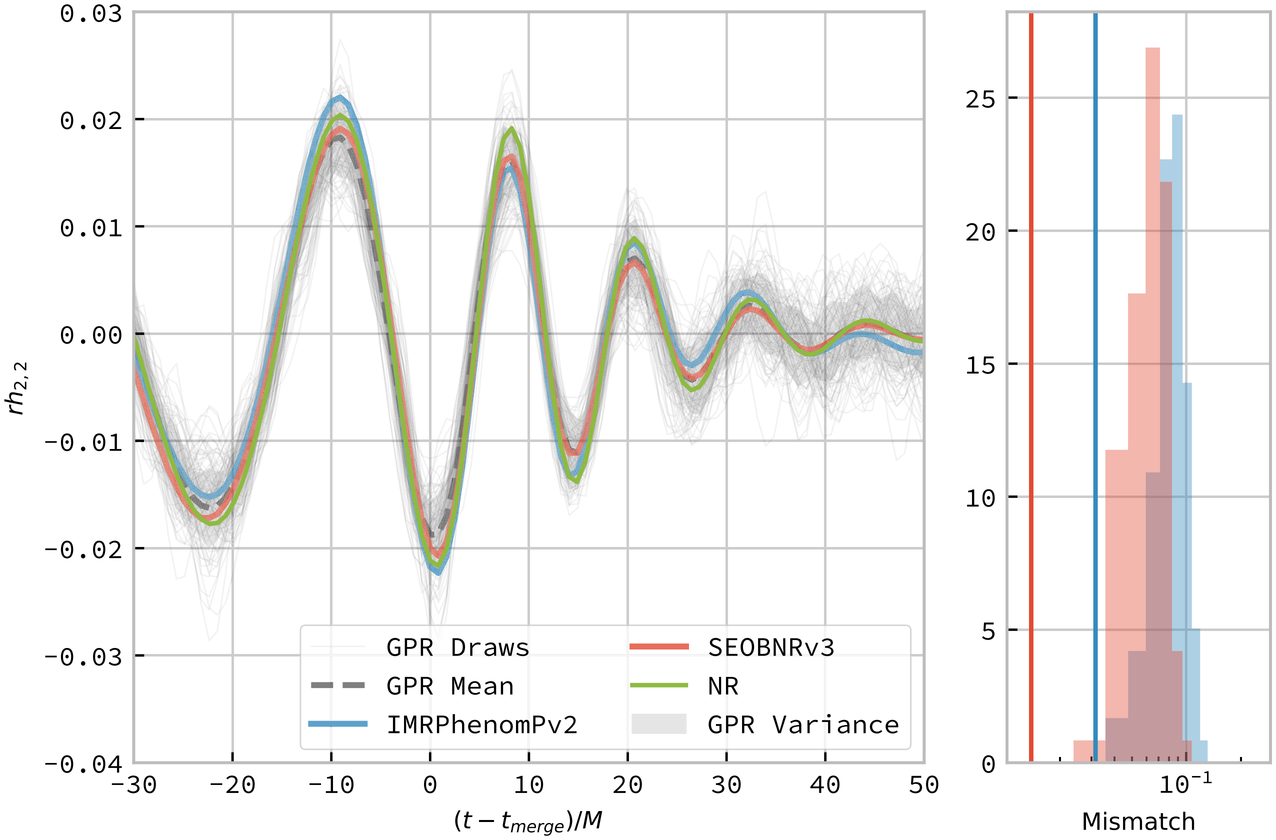

FIG. 7. GPR predictions, compared to NR. One hundred draws from the Gaussian process (left panel) for a nonspinning configuration ($s_1 = (0, 0, 0)$, $s_2 = (0, 0, 0)$, $q = 0.625$), shown as light gray lines compared to the phenomenological approximant models, IMRPhenomPv2 in blue and SEOBNRv3 in red. The mean draw from the Gaussian process is shown as a gray dashed line, while the associated variance is plotted as a gray-filled region surrounding the mean. The differences between the phenomenological model and the GPR model waveforms are seen to also exist between the phenomenological model waveforms and the NR-derived waveform, plotted here in purple. In the right panel, the distribution of mismatches between the samples and both phenomenological waveforms is shown, with the vertical lines representing the mismatch between the mean waveform from the GPR and each phenomenological waveform.

computational burden of evaluating the model, a hierarchical matrix inversion method was used (described in [29] and discussed in Sec. III B).

We present three testing strategies for our GPR model, in addition to a number of waveforms which have been produced by it in Sec. IV. We present both the results of these tests and make comparisons between the model's output and two well-established phenomenological models. This difference also occurs between the phenomenological model and the waveform produced from NR. A number of phenomena are likely to have contributed to this discrepancy. One such difference is in the systematic errors of the NR simulations used to produce the training data for the GPR model compared to those used to calibrate the phenomenological models. Additionally, the relatively small number of waveforms used to calibrate the phenomenological models compared to the GPR model are likely to introduce systematic errors in the waveforms produced by those models. In order to reduce the effect of systematic errors from NR, a larger model could include waveforms from a number of different NR waveform catalogs; however, the addition of more waveforms will increase the memory requirements to both train and evaluate the model. Our waveform model tends toward producing conservative estimates of the waveform; this is clearly visible in the variance of the precessing waveform in Fig. 6. The use of additional waveforms is likely to improve the confidence of the model's prediction.

In order for a GPR-based approach such as this to be practical for parameter estimation studies using data from LIGO or Virgo, it would be necessary to have a means of producing waveforms which are capable of modeling a greater amount of the inspiral than our model can currently provide. One potential approach to solving this problem is hybridizing the output waveform from our GPR model with waveforms produced from a post-Newtonian approximant, in a similar manner to that used by [21]. This would allow us to overcome the need for much longer waveforms to be used in the training set, while still allowing the production of waveforms with lengthier inspirals than our model is currently capable of.

We note that in this work we have not attempted to benchmark this model and compare the times required to produce sample waveforms from it compared to the analytical approximates which are currently in regular use. We expect to address this shortcoming in future work, but acknowledge that a number of optimizations may be made to allow the model to produce results more expediently without impacting on its accuracy.

In conclusion, we have demonstrated that GPR is capable of being used as an interpolant for BBH waveforms, trained directly off data from NR simulations. While this method cannot hope to produce waveforms with the same precision as NR itself, it does account for the uncertainty introduced through interpolation, a feature which is valuable for preventing the introduction of bias in a PE analysis.

ACKNOWLEDGMENTS

The authors wish to thank Christopher Moore, Sebastian Khan, and Vijay Varma for their insightful comments and suggestions on an earlier draft of the paper and also our two anonymous reviewers for their suggestions. D. W. received support from Science and Technology Facilities Council (STFC) Grants No. ST/N504075/1 and No. ST/L000946/1. I. S. H. is supported by STFC Grant No. ST/L000946/1. B. K. acknowledges support from NSF Awards No. PHY-1806580, No. PHY-1809572, and No. 1333360. This document has been assigned LIGO document reference LIGO-P1800128. The software described in this paper is available on GitHub (https://www.github.com/transientlunatic/heron) and on Zenodo [30]. The analysis and software development described in this paper was made possible thanks to a number of open source PYTHON libraries, including NUMPY [31], MATPLOTLIB [32], and GEORGE [29].

[1] B. Szilagyi, J. Blackman, A. Buonanno, A. Taracchini, H. P. Pfeiffer, M. A. Scheel, T. Chu, L. E. Kidder, and Y. Pan, Phys. Rev. Lett. **115**, 031102 (2015).

[2] C. Devine, Z. B. Etienne, and S. T. McWilliams, Classical Quantum Gravity **33**, 125025 (2016).

[3] P. Ajith, S. Babak, Y. Chen, M. Hewitson, B. Krishnan, A. M. Sintes, J. T. Whelan, B. Brügmann, P. Diener, N. Dorband, J. Gonzalez, M. Hannam, S. Husa, D. Pollney, L. Rezzolla, L. Santamaría, U. Sperhake, and J. Thornburg, Phys. Rev. D **77**, 104017 (2008).

[4] M. Hannam, P. Schmidt, A. Bohé, L. Haegel, S. Husa, F. Ohme, G. Pratten, and M. Pürrer, Phys. Rev. Lett. **113**, 151101 (2014).

[5] S. Khan, K. Chatziioannou, M. Hannam, and F. Ohme, Phys. Rev. D **100**, 024059 (2019).

[6] S. A. Usman *et al.*, Classical Quantum Gravity **33**, 215004 (2016).

[7] A. Nitz *et al.*, ligo-cbc/pycbc: O2 production release 20 (2017), https://doi.org/10.5281/zenodo.883086.

[8] T. Dal Canton, A. H. Nitz, A. P. Lundgren, A. B. Nielsen, D. A. Brown, T. Dent, I. W. Harry, B. Krishnan, A. J. Miller, K. Wette, K. Wiesner, and J. L. Willis, Phys. Rev. D **90**, 082004 (2014).

[9] A. Buonanno and T. Damour, Phys. Rev. D **59**, 084006 (1999).

[10] A. Buonanno and T. Damour, Phys. Rev. D **62**, 064015 (2000).

[11] T. Damour and A. Nagar, arXiv:0906.1769.

[12] Y. Pan, A. Buonanno, M. Boyle, L. T. Buchman, L. E. Kidder, H. P. Pfeiffer, and M. A. Scheel, Phys. Rev. D **84**, 124052 (2011).

[13] A. Buonanno, Y. Pan, J. G. Baker, J. Centrella, B. J. Kelly, S. T. McWilliams, and J. R. van Meter, Phys. Rev. D **76**, 104049 (2007).

[14] A. Taracchini, Y. Pan, A. Buonanno, E. Barausse, M. Boyle, T. Chu, G. Lovelace, H. P. Pfeiffer, and M. A. Scheel, Phys. Rev. D **86**, 024011 (2012).

[15] A. Taracchini, A. Buonanno, Y. Pan, T. Hinderer, M. Boyle, D. A. Hemberger, L. E. Kidder, G. Lovelace, A. H. Mroué, H. P. Pfeiffer, M. A. Scheel, B. Szilágyi, N. W. Taylor, and A. Zenginoglu, Phys. Rev. D **89**, 061502 (2014).

[16] A. Bohé, L. Shao, A. Taracchini, A. Buonanno, S. Babak, I. W. Harry, I. Hinder, S. Ossokine, M. Pürrer, V. Raymond, T. Chu, H. Fong, P. Kumar, H. P. Pfeiffer, M. Boyle, D. A. Hemberger, L. E. Kidder, G. Lovelace, M. A. Scheel, and B. Szilágyi, Phys. Rev. D **95**, 044028 (2017).

[17] Y. Pan, A. Buonanno, A. Taracchini, L. E. Kidder, A. H. Mroué, H. P. Pfeiffer, M. A. Scheel, and B. Szilágyi, Phys. Rev. D **89**, 084006 (2014).

[18] J. Blackman, S. E. Field, C. R. Galley, B. Szilágyi, M. A. Scheel, M. Tiglio, and D. A. Hemberger, Phys. Rev. Lett. **115**, 121102 (2015).

[19] J. Blackman, S. E. Field, M. A. Scheel, C. R. Galley, D. A. Hemberger, P. Schmidt, and R. Smith, Phys. Rev. D **95**, 104023 (2017).

[20] J. Blackman, S. E. Field, M. A. Scheel, C. R. Galley, C. D. Ott, M. Boyle, L. E. Kidder, H. P. Pfeiffer, and B. Szilágyi, Phys. Rev. D **96**, 024058 (2017).

[21] V. Varma, S. E. Field, M. A. Scheel, J. Blackman, L. E. Kidder, and H. P. Pfeiffer, Phys. Rev. D **99**, 064045 (2019).

[22] C. J. Moore and J. R. Gair, Phys. Rev. Lett. **113**, 251101 (2014).

[23] C. J. Moore, C. P. L. Berry, A. J. K. Chua, and J. R. Gair, Phys. Rev. D **93**, 064001 (2016).

[24] Z. Doctor, B. Farr, D. E. Holz, and M. Pürrer, Phys. Rev. D **96**, 123011 (2017).

[25] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)* (MIT Press, Cambridge, MA, 2005).

[26] D. P. Kingma and J. Ba, arXiv:1412.6980.

[27] K. Jani, J. Healy, J. A. Clark, L. London, P. Laguna, and D. Shoemaker, Classical Quantum Gravity **33**, 204001 (2016).

[28] P. Schmidt, I. W. Harry, and H. P. Pfeiffer, arXiv:1703.01076.

[29] S. Ambikasaran, D. Foreman-Mackey, L. Greengard, D. W. Hogg, and M. O'Neil, IEEE **38**, 252 (2015).

[30] D. Williams, transientlunatic/heron: Castle Semple (2019), https://doi.org/10.5281/zenodo.3378679.

[31] S. van der Walt, S. C. Colbert, and G. Varoquaux, Comput. Sci. Eng. **13**, 22 (2011).

[32] J. D. Hunter, Comput. Sci. Eng. **9**, 90 (2007).