

BayesWave analysis pipeline in the era of gravitational wave observationsNeil J. Cornish¹,[✉] Tyson B. Littenberg²,[✉] Bence Bécsey,¹ Katerina Chatziioannou,³ James A. Clark,⁴ Sudarshan Ghonge⁴,[✉] and Margaret Millhouse⁵¹*eXtreme Gravity Institute, Department of Physics, Montana State University, Bozeman, Montana 59717, USA*²*NASA Marshall Space Flight Center, Huntsville, Alabama 35812, USA*³*Center for Computational Astrophysics, Flatiron Institute, 162 5th Avenue, New York, New York 10010, USA*⁴*Center for Relativistic Astrophysics and School of Physics, Georgia Institute of Technology, Atlanta, Georgia 30332, USA*⁵*OzGrav, University of Melbourne, Parkville, Victoria 3010, Australia*

(Received 19 November 2020; accepted 14 January 2021; published 2 February 2021)

We describe updates and improvements to the BayesWave gravitational wave transient analysis pipeline, and provide examples of how the algorithm is used to analyze data from ground-based gravitational wave detectors. BayesWave models gravitational wave signals in a morphology-independent manner through a sum of frame functions, such as Morlet-Gabor wavelets or chirplets. BayesWave models the instrument noise using a combination of a parametrized Gaussian noise component and nonstationary and non-Gaussian noise transients. Both the signal model and noise model employ trans-dimensional sampling, with the complexity of the model adapting to the requirements of the data. The flexibility of the algorithm makes it suitable for a variety of analyses, including reconstructing generic unmodeled signals; cross-checks against modeled analyses for compact binaries; as well as separating coherent signals from incoherent instrumental noise transients (glitches). The BayesWave model has been extended to account for gravitational wave signals with generic polarization content and the simultaneous presence of signals and glitches in the data. We describe updates in the BayesWave prior distributions, sampling proposals, and burn-in stage that provide significantly improved sampling efficiency. We present standard review checks indicating the robustness and convergence of the BayesWave trans-dimensional sampler.

DOI: [10.1103/PhysRevD.103.044006](https://doi.org/10.1103/PhysRevD.103.044006)**I. INTRODUCTION**

The era of gravitational wave observations began in earnest in September 2015 with the first detection of gravitational waves from a binary black hole merger [1]. In anticipation of the detections, a new approach to gravitational wave data analysis was proposed [2,3] that uses trans-dimensional Bayesian inference to model the instrument noise and short duration gravitational wave signals of arbitrary morphology. The foundational principle behind this approach is to allow the complexity of the model to automatically adapt according to the complexity of the data, following the motto “model everything and let the data sort it out.” This approach was implemented in the BayesWave algorithm [2], which models gravitational wave transients and noise transients as a collection of continuous wavelets, and the BayesLine algorithm [3], which models the power spectral density of the instrument noise using a smooth spline model and a collection of Lorentzian lines. Since the BayesLine algorithm is a key component and fully integrated in the BayesWave algorithm, we will collectively refer to them as BayesWave going forward.

The BayesWave algorithm has been used extensively in the analysis of data from the LIGO [4] and Virgo [5] gravitational wave detectors. Applications include template-free reconstructions of gravitational wave signals, residual based tests of general relativity, noise transient removal and power spectral density estimation for parameter estimation studies. In the intervening years, the BayesWave algorithm has undergone changes, with the addition of new functionality and improvements in the sampling efficiency, and computational cost. This paper serves as an update to BayesWave, consistent with the publicly available software [6]. Section II serves as an overview of the BayesWave algorithm and use cases. Updates are organized by changes to the underlying data model used by BayesWave in Sec. III, and changes to the stochastic sampling engine which improve convergence are described in Sec. IV. Section V describes the inherent postprocessing steps, while Sec. VI discusses standard review tests for samplers. Section VII concludes by outlining future development and use cases. Appendices A, B, and C go into details about the deployment, performance, and optimizations, respectively.

II. OVERVIEW OF BAYESWAVE

The foundation of the BayesWave algorithm is a model for the data from detector I : $d_I = h_I + g_I + n_I$, where h_I is the detector's response to a gravitational wave signal, g_I are non-Gaussian noise transients, or “glitches” in the data, and n_I is the random detector noise. The transients h and g are modeled using coherent parametrized fits to the waveforms $h_I^M(\theta_h)$ and $g_I^M(\theta_{g,I})$, with superscript M indicating that it is a model of the true signal, while n_I is modeled statistically with a parametrized noise covariance matrix $C_I(\theta_n)$. The model parameters are optimized using a Markov chain Monte Carlo (MCMC) sampler of the posterior distribution function

$$p(\theta|d) = \frac{p(d|\theta)p(\theta)}{p(d)}, \quad (1)$$

where the joint parameter set is $\theta = [\theta_h, \theta_{g,I}, \dots, \theta_{g,K}, \theta_{n,I}, \dots, \theta_{n,K}]$ for detectors $I, \dots, K$, and the joint data are $\mathbf{d} = [d_I, \dots, d_K]$ and similarly for the signal $\mathbf{h}^M$ and glitch $\mathbf{g}^M$ models. The likelihood for the Fourier domain residual $\tilde{\mathbf{r}} = \tilde{\mathbf{d}} - \tilde{\mathbf{h}}^M - \tilde{\mathbf{g}}^M$, assuming that the remaining noise is Gaussian distributed, is

$$p(d|\theta) = \frac{1}{\det(\pi\mathbf{C})} e^{-\tilde{\mathbf{r}}^* \mathbf{C}^{-1} \tilde{\mathbf{r}}} \quad (2)$$

and $p(\theta)$ and $p(d)$ are the prior and marginalized likelihood, respectively.

When data from multiple detectors are considered the noise is assumed to be independent between detectors so the full noise covariance matrix $\mathbf{C}$ is block diagonal. Furthermore, the variance of the noise is assumed to be constant over the observation period (stationary) of duration T , and thus each block of $\mathbf{C}$ is itself diagonal with the only nonzero elements being proportional to the variance $\langle n_{[i,j]} n_{[k,l]}^* \rangle = \frac{T}{2} S_{n,[i,j]} \delta_{i,k} \delta_{j,l}$, where $S_{n,[i,j]}$ is the noise power spectral density S_n of detector i in data sample j . Under these assumptions, the likelihood in Eq. (2) reduces to

$$p(d|\theta) = \prod_i \prod_j \frac{2}{\pi T S_{n,[i,j]}} e^{-\frac{2|\tilde{r}_{[i,j]}|^2}{T S_{n,[i,j]}}} \quad (3)$$

again with indices i spanning the data streams and j the data samples. See Ref. [7] for details on the likelihood derivation for discretely sampled data in the Fourier domain.

As originally described in Ref. [2], both the signal and the glitch models are constructed from a linear combination of sine-Gaussian wavelets

$$\begin{aligned} \Psi(t; \vec{\lambda}) &= A e^{(t-t_0)^2/\tau^2} \cos(2\pi f_0(t-t_0) + \phi_0) \\ \tilde{\Psi}(f; \vec{\lambda}) &= \frac{\sqrt{\pi} A \tau}{2} e^{-\pi^2 \tau^2 (f-f_0)^2} (e^{i(2\pi(f-f_0)t_0 + \phi_0)} \\ &\quad + e^{-i(2\pi(f+f_0)t_0 + \phi_0)} e^{-Q^2 f/f_0}) \end{aligned} \quad (4)$$

with $\vec{\lambda} \rightarrow (t_0, f_0, Q, A, \phi_0)$, where t_0 is the central time of the wavelet, f_0 is frequency at $t = t_0$, Q is the wavelet quality factor (i.e., the number of cycles of the wavelet over one e -folding of the Gaussian envelope), $\tau = Q/2\pi f_0$, A is the wavelet amplitude, and ϕ_0 is the wavelet phase at $t = t_0$. The wavelets form a frame, not a basis, since they are not linearly independent. The glitch model wavelet parameters are independent in each detector, while the signal model wavelets are coherently projected onto each detector using a set of extrinsic parameters (see Sec. III A). In the standard configuration, BayesWave assumes that the gravitational wave signal is elliptically polarized, as is the case for quasicircular, nonprecessing compact binary coalescences (CBCs). In that case there are four extrinsic parameters: sky location angles specifying the right ascension α and declination δ , the polarization angle ψ , and the ellipticity parameter ϵ which maps the $+$ polarization to the $\times$ polarization via $h_\times = \epsilon h_+ e^{i\pi/2}$. The total number of wavelets used in the glitch and signal model is marginalized over with a trans-dimensional MCMC, making the number of wavelets in each model a free parameter of the model.

BayesWave was designed with a generic application programming interface for the wavelet functions. In principle, the sine-Gaussian wavelets are easily replaced by any other set of frame functions. There is currently one alternative wavelet model available in BayesWave. Reference [8] describes the inclusion of a *chirplet* frame which is similar to the sine-Gaussian wavelets but modified with a constant frequency derivative $\dot{f}_0$. The added flexibility of the chirplet model to track rapidly changing frequency content of a signal allows for better fits to certain waveform morphologies.

The Gaussian noise model is handled by an independently developed MCMC algorithm, BayesLine [3], although the name BayesWave has become synonymous for both algorithms. In BayesLine, the Gaussian noise model is decomposed into two components evident in LIGO and Virgo data—broad spectrum and gradually varying noise due to e.g., ground motion, thermal fluctuations in the optics, and shot noise from the interferometric sensing; and narrow band, high amplitude, spectral lines due to mechanical resonances in the detectors, the power supply at detector, and calibration lines intentionally added to the data. The broadband noise is modeled as a cubic splines interpolation between control points parametrized by their frequency and noise level, the number and location of which are adjustable by the sampler. The narrow band noise is modeled with a linear combination of Lorentzian-like functions parametrized by the central frequency, width, and line height. Again, a trans-dimensional MCMC algorithm is used to marginalize over the number of Lorentzians used in the fit. BayesLine has been shown to outperform periodogram-based approaches for spectral estimation in LIGO-Virgo data because it only assumes the noise to be stationary over the interval of data being analyzed, as

opposed to the interval of data needed for the periodogram [9].

The combined BayesWave algorithm uses a blocked Gibbs sampler, alternating between updates to the Gaussian noise model, the intrinsic parameters which control the number and shape of the wavelet model, and (for the signal model) the extrinsic parameters which govern the coherent projection of the model onto the network of detectors.

To evaluate different hypothesis about the data, BayesWave is run in a restricted setting only allowing certain models in the fit and then using thermodynamic integration [10] to compute the evidence for the model. The standard workflow includes assessing a Gaussian noise-only model, a model containing Gaussian noise and the joint glitch model for all detectors, and the Gaussian noise plus signal model which requires *at least one wavelet* to be coherently projected onto the network. The Bayes factor between Gaussian noise plus signal model and the Gaussian noise plus glitch model is a robust detection statistic [11].

A. Existing proposals and priors

Priors.—The BayesWave default behavior is to use flat priors on all intrinsic parameters except the wavelet amplitude A , and for all extrinsic parameters. For intrinsic parameters, the prior ranges cover: $t_0 \in U[t_{\min}, t_{\max}]$, where by default $t_{\min}$ and $t_{\max}$ enclose a 1 s interval centered on the candidate gravitation wave (GW) event time, though the interval and location in the full data segment are adjustable by the user; $f_0 \in U[f_{\min}, f_{\max}]$, where $f_{\min}$ is specified by the user and $f_{\max}$ is the Nyquist frequency determined from the user-requested sampling rate for the input data; $Q \in U[0.1, 40]$ by default but is adjustable by the user; $\phi_0 \in U[0, 2\pi]$. For signal-model extrinsic parameters, the ranges $\alpha \in U[0, 2\pi]$; $\sin \delta \in U[-1, 1]$ which, combined with the prior on α make the joint prior uniform on the sky; $\psi \in U[0, \pi]$; $\epsilon \in U[-1, 1]$; and an overall phase applied to all wavelets in the signal model $\varphi \in U[0, 2\pi]$. This overall phase shift is degenerate with a simultaneous shift of all wavelet phases ϕ_0 by the same amount, but the explicit inclusion of φ aids the sampler convergence.

The amplitude prior is based on the signal-to-noise ratio, ρ , of each wavelet and is designed to suppress low amplitude wavelets which will not contribute to the likelihood while also not biasing the amplitude recovery of high ρ signals. The priors take slightly different functional form for the glitch and signal model to account for the fact that high amplitude glitches ($\rho > 100$) are not uncommon in the data whereas GW signals do not reach such levels at the current detector sensitivities. More detail on the amplitude prior is provided in Sec. III C 1.

By default BayesWave also uses flat priors on the number of wavelets $D \in [D_{\min}, 100]$ though the maximum number of wavelets is adjustable by the user, with $D_{\min} = 1(0)$ for the signal (glitch) model. Note that nested within

the glitch model is the Gaussian noise model, however it is sometimes worth evaluating the Gaussian-noise only model by itself as the posterior weight of the glitch model in the zero wavelet case may be impractically small given the number of posterior samples in the chain.

Proposals.—BayesWave uses a mixture of different proposal distributions for generating trial parameters. These include fair draws from the prior to ensure efficient sampling of the full parameter space for the high temperature chains. Draws from the prior are used for within-model and transdimensional proposed moves. BayesWave also relies on custom-made proposals that leverage what is known, or can easily be inferred, from the data or model. In addition to the new developments described in Sec. IV, the BayesWave sampler particularly benefits from proposals along eigenvectors, scaled by the eigenvalues, of the Fisher information matrix approximation to the inverse covariance matrix $C_{ij}^{-1} \sim \Gamma_{ij} \equiv (\frac{\partial \Psi}{\partial \lambda_i} | \frac{\partial \Psi}{\partial \lambda_j})$. The elements of the Fisher matrix for the wavelets are known analytically, while matrix elements for the extrinsic parameters of the signal model are computed numerically.

BayesWave also uses proposals that encourage placing new wavelets near in time frequency to existing groups of wavelets. The proposal is built by summing Gaussians at existing wavelet locations in such a way that regions in the time-frequency plane near to, but not overlapping, with existing wavelets are preferentially tried by the sampler. All of the aforementioned proposals are explained in Ref. [2].

One final proposal worth noting here takes advantage of a near degeneracy between the reference time and reference phase of a wavelet, particularly for wavelets with a high Q parameter. The degeneracy arises because the wavelets can be time shifted by a cycle and still match a high- Q feature in the data by compensating by adjusting the wavelet phase. This proposal is described in detail in Ref. [12].

B. Use cases

The flexibility of the BayesWave signal, glitch, and noise models to adapt to features in the data, coupled with the restraint on the models applied by comparing evidences, have made the algorithm well suited to a broad range of LIGO-Virgo analyses as a tool to study both the signals and the noise in the data. On the search side, BayesWave is part of a hierarchical detection pipeline for generic short-duration gravitational wave transients (i.e., “bursts”) [15–13]]. Regarding glitches in data that contain a candidate signal, BayesWave was adapted as a data cleaning tool to subtract the noise transient that overlapped with GW170817 [16], while preserving the fidelity of the underlying signal analysis [17]. Extensive glitch subtraction was subsequently part of the GWTC-2 catalog [18]. On the noise side, BayesWave is used as a spectral estimation tool that provides the noise model to the template-based parameter estimation follow-up of compact binary signals [9,19].

Regarding analysis of gravitational wave signals, BayesWave has been used to study unmodeled or poorly modeled sources, such as the postmerger emission from neutron star binaries [20–22], eccentric black hole binaries [23], postmerger echoes from black hole binaries [24,25], and supernova [26]. Additionally, it has contributed waveform reconstructions and inferences of candidate GW transients without the use of templates. BayesWave’s generic reconstructions are compared to the template-based reconstructions as part of event validation studies [1,19,27,28]. Related to this, BayesWave has been used to search for excess power in residual data obtained after the subtraction of a template-based point estimate to signals, thus testing how well the physically motivated waveform models match the data; this analysis has been interpreted as a model-agnostic test of general relativity [29–31].

III. MODEL EXTENSIONS

In this section we describe the new BayesWave model capabilities, including changes in the signal and glitch models, as well as new supported priors.

A. Signal polarization

In general relativity, GWs contain two polarization modes, colloquially referred to as plus and cross. GW detectors respond differently to each of these modes, encoded in the detector antenna pattern functions $F_{\times}(\Omega, \psi)$, $F_{+}(\Omega, \psi)$, where $\vec{\Omega} \rightarrow (\alpha, \delta)$ describes the sky location of the source and ψ is the polarization angle. The response of a detector to an impinging signal can then be expressed as

$$h_I(f) = (F_{\times}(\Omega, \psi)h_{\times}(f) + F_{+}(\Omega, \psi)h_{+}(f))e^{2\pi i f \Delta t(\Omega)}, \quad (5)$$

where $h_I(f)$ is the interferometric response, $\Delta t(\Omega)$ is the light travel time from a fiducial reference location to the detector, and $h_{\times}(f)$, $h_{+}(f)$ are the cross and plus signal respectively expressed at the reference location. BayesWave uses an arbitrarily chosen detector as the reference location.

The projection from the reference location to each individual detector in the network, Eq. (5), contains not only the two GW polarization modes, but also the sky location and orientation. This means that with two or three detectors available, the problem of extracting GW polarizations from the observed detector response might be underconstrained. For this reason, the original signal model in BayesWave restricted the polarization content of the signal to the case of elliptical polarization

$$\begin{aligned} h_{+} &= \sum_n \Psi(f; t_0^n, f_0^n, Q^n, \mathcal{A}^n, \phi_0^n), \\ h_{\times} &= i\epsilon h_{+}, \end{aligned} \quad (6)$$

where ϵ is the *ellipticity* parameter encoding the degree of elliptical polarization. For $\epsilon = 0$, $h_{\times} = 0$ and the signal is linearly polarized; if $\epsilon = 1$ the signal is circularly polarized. This assumption of elliptical polarization does not hold for complicated CBC signals whose polarization content changes with time, such as spin-precessing signals or signals with a strong higher-order modes content. Additionally, generic bursts of GWs are not expected to possess any special polarization content. Previous studies have shown that restricting the signal model to elliptical polarization is suboptimal for detecting unpolarized signals such as white noise bursts [12].

In order to relax the elliptical polarization constraint, we generalize the signal model in BayesWave to

$$\begin{aligned} h_{+} &= \sum_n \Psi(f; t_0^n, f_0^n, Q^n, \mathcal{A}^{n,+}, \phi_0^{n,+}), \\ h_{\times} &= \sum_n \Psi(f; t_0^n, f_0^n, Q^n, \mathcal{A}^{n,\times}, \phi_0^{n,\times}), \end{aligned} \quad (7)$$

while setting $\psi = 0$ in Eq. (5). This generic polarization model assumes that each polarization state can be expressed as a sum of the same number n of wavelets that have the same quality factor Q^n , central time t_0^n , and central frequency f_0^n , but differ in amplitude and phase. We argue that we can restrict the plus and cross wavelets to the same (Q, t_0, f_0) without loss of generality. From Eq. (5) it is clear that the time-frequency content of the GW signal is independent of the detector network, and only its amplitude and phase are modified by the process of projecting it from the reference location onto a detector network. This has the added benefit of avoiding pathological solutions where all of h_{+} is in one detector and all of $h_{\times}$, making it degenerate with the glitch model.

We showcase the generic polarization model by analyzing simulated CBC signals observed by a network of Hanford, Livingston and Virgo (HLV) GW detectors at design sensitivity with a network signal-to-noise ratio $\rho = 100$. To simulate the observed data we use the waveform model IMRPhenomPv2 [32] and assume a zero noise realization. The component masses are set to $m_1 = 20 M_{\odot}$, $m_2 = 5 M_{\odot}$, as unequal masses are known to maximize the effect of precession, and hence deviation from elliptical polarization [33,34]. Besides the system’s mass ratio, the binary inclination and the amount of in-plane spin also affect the degree to which a signal is precession modulated. We employ two values of the inclination angle between the line of sight and the orbital angular momentum $\iota = \{45^\circ, 90^\circ\}$, defined at $f = 16$ Hz. The inclination angle evolves under spin precession, so an originally edge-on system ($\iota = 90^\circ$) will not remain in this configuration. The in-plane spin is commonly characterized through the χ_p parameter [35] and we inject signals with $\chi_p = \{0, 0.52, 0.98\}$, again defined at 16 Hz. The first case corresponds to a spin-aligned system, studied for reference. The value of

χ_p also evolves under spin precession, and it is not directly related to how prominent precessional modulations are in the observed signal [33].

We show reconstructions of the observed data in the Livingston detector in Fig. 1 for different values of χ_p . Shaded regions show the 90% credible interval for the

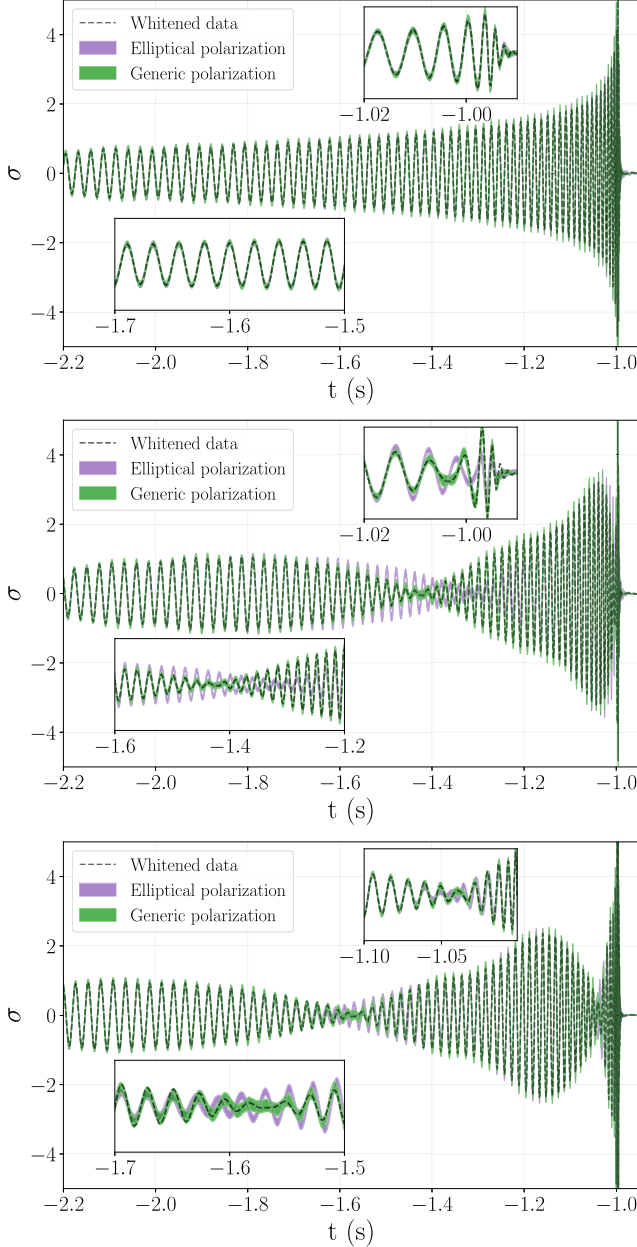


FIG. 1. Whitened time-domain reconstructions of injected GW signals. We plot the injected data in gray dashed lines, where no additional noise realization has been added. Purple (green) shaded regions show the 90% credible interval for the reconstruction when assuming an elliptical (generic) polarization content for the signal. From top to bottom we have signals with network signal-to-noise ratio $\rho = 100$ and with $(\chi_p = 0, \iota = 45^\circ)$, $(\chi_p = 0.52, \iota = 45^\circ)$, $(\chi_p = 0.98, \iota = 90^\circ)$.

reconstruction when assuming an elliptical polarization (purple) and a generic polarization (green). The top panel contains a signal with $\chi_p = 0$, which exhibits no precessional modulations. Both analyses reconstruct the signal similarly well. The middle and bottom panels show the results for $\chi_p = 0.52$ and $\chi_p = 0.98$, respectively. In both cases we find that the BayesWave analysis that allows for a generic signal polarization does a better job of reconstructing the injected signal. Despite this improvement, the elliptical polarization analysis is still able to reproduce the spin-induced amplitude modulation to some extent, suggesting that signals of extreme ρ and precession are needed before the elliptical polarization approximation results in considerably deteriorated inference. Interestingly, we also find that the elliptical polarization analysis performs better for the $\chi_p = 0.98$ signal than the $\chi_p = 0.52$ one, again suggesting that χ_p might not be a suitable parameter to quantify the amount of spin precession present [33].

Besides more faithful signal reconstruction, the generic polarization analysis also allows us to infer the polarization content of the observe signal. We employ the usual Stokes parameters [7], defined as

$$U = \tilde{h}_+ \tilde{h}_\times^* + \tilde{h}_\times \tilde{h}_+^*, \quad (8)$$

$$V = i(\tilde{h}_+ \tilde{h}_\times^* - \tilde{h}_\times \tilde{h}_+^*), \quad (9)$$

$$I = |\tilde{h}_+|^2 + |\tilde{h}_\times|^2, \quad (10)$$

$$Q = |\tilde{h}_+|^2 - |\tilde{h}_\times|^2, \quad (11)$$

which here are to be understood as being a function of the GW frequency. For an elliptically polarized signal with ellipticity ϵ these reduce to $U \sim 0, (I - Q)/(I + Q) \sim \epsilon^2, V/(I + Q) \sim \epsilon, (I - Q)/V \sim \epsilon$. Figure 2 shows these combinations as a function of frequency for the three values of χ_p studied and $\iota = 45^\circ$. The dashed black lines show the injected values computed directly from the simulated signal, while shaded regions show the 90% credible interval for the reconstruction when employing the generic polarization analysis. In all cases the generic polarization model is able to reconstruct the stokes parameters and their frequency evolution, suggesting that the generic analysis can be used to reconstruct the polarization content of a detected signal in a morphology-agnostic way.

B. Signal plus glitch model

In the original version of BayesWave the signal and glitch models were disjoint hypotheses to be tested, where the former assumed that all wavelets in the model were coherent across the detector network, and the latter assumed that all wavelets were independent in each detector. We have now added a joint hypothesis where the data can contain both a coherent signal *and* additional

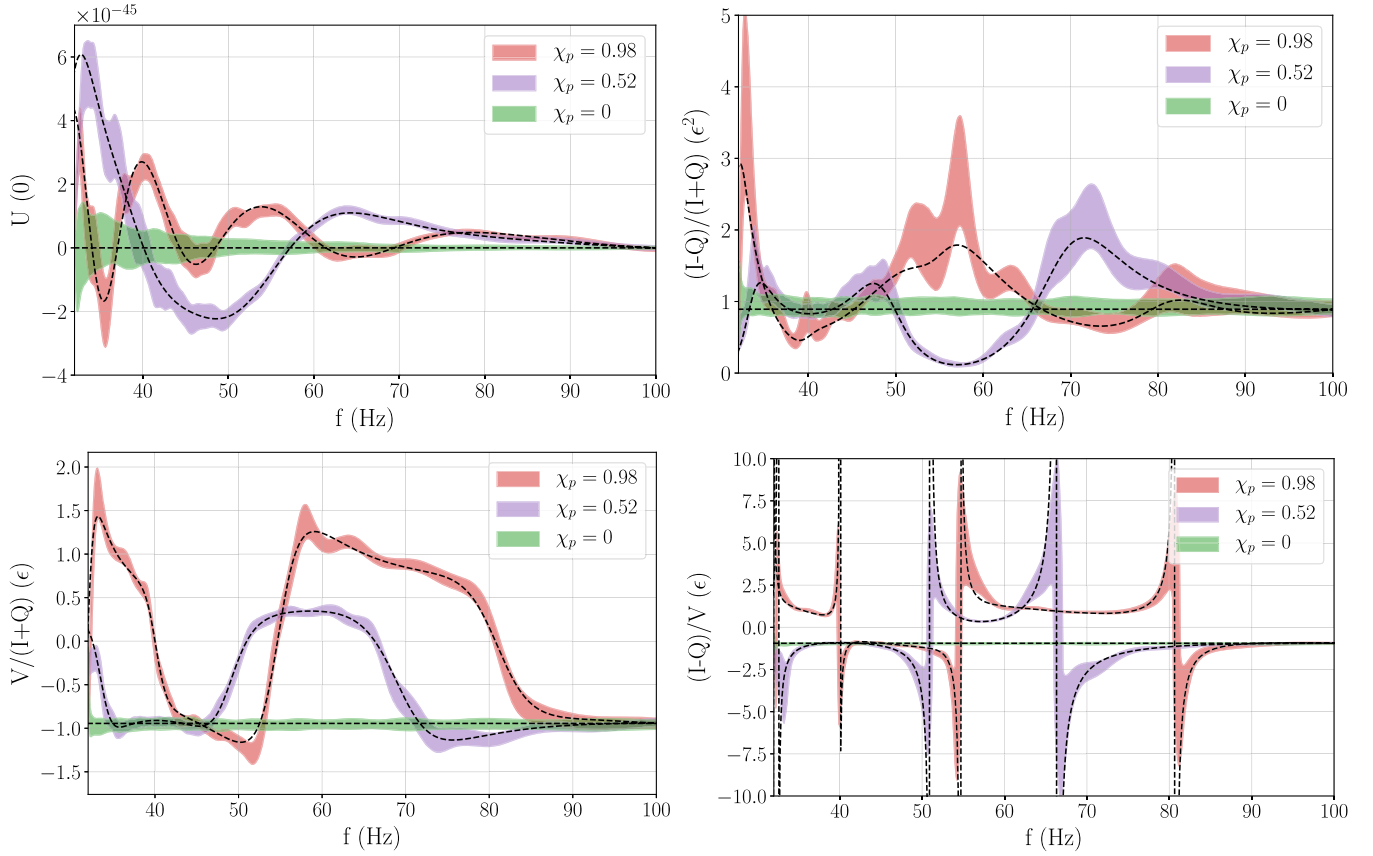


FIG. 2. Stokes parameter combinations as a function of frequency for the three injections at $\rho = 100$ and $\iota = 45^\circ$. Dashed lines show the injected values and shaded regions show the 90% credible interval obtained under the generic polarization analysis. The parentheses on the y axis label indicates the constant value each Stokes combination assumes under elliptical polarization.

glitches in any detector's data (S + G). The new model is made possible by the improved mixing of the Markov chain from the changes described in Sec. IV.

The joint model is of particular value when trying to identify and mitigate glitches that occur near a candidate GW signal, in which case the glitch model can be used to remove the excess noise from the data. Using BayesWave to remove glitches from data near GW candidates was first used in the analysis of GW170817 [16,17], when the part of the GW signal detectable by BayesWave was sufficiently far from the glitch time that there was no concern of the wavelet model also removing some signal power. Using the S + G model improves the “safety” of the glitch subtraction as any coherent features in the data will be picked up by the signal model and only the excess power independent in either detector will be fit and removed by the glitch model.

Figure 3 demonstrates the S + G model on data from the first observing run (O1) containing a common glitch type in one detector. In this example the glitch was in the Hanford detector, but the glitch type is common in both Hanford and Livingston [36]. A simulated binary black hole (BBH) signal with parameters similar to those of GW150914 was added to the data with merger time just before, coincident, and just after the glitch time. BayesWave was used to

process the data with the BBH signal added using the S + G model, and we compare the signal reconstructions (top panel) and the glitch reconstructions (bottom panel) for each injection. In each case the signal and glitch recovery is self-consistent, regardless of how much the glitch and signal overlap in time. The glitch reconstructions are further compared to a glitch-only analysis of the original data with no signal added (dark gray) again showing that the glitch recovery, and therefore the subsequent glitch subtraction, is robust.

Histograms of the number of wavelets used by the signal and glitch model for each analysis are shown in Fig. 4. The distribution of signal and glitch wavelets used is largely the same, no matter where the signal is injected relative to the glitch. The case where no signal is injected, indicated by gray bars in Fig. 4, shows the same distribution of glitch model wavelets, but essentially no support for adding signal wavelets.

C. Updated priors

1. Wavelet amplitude prior

In both the signal and the glitch model, the prior on the amplitudes of the individual wavelets is actually given by a

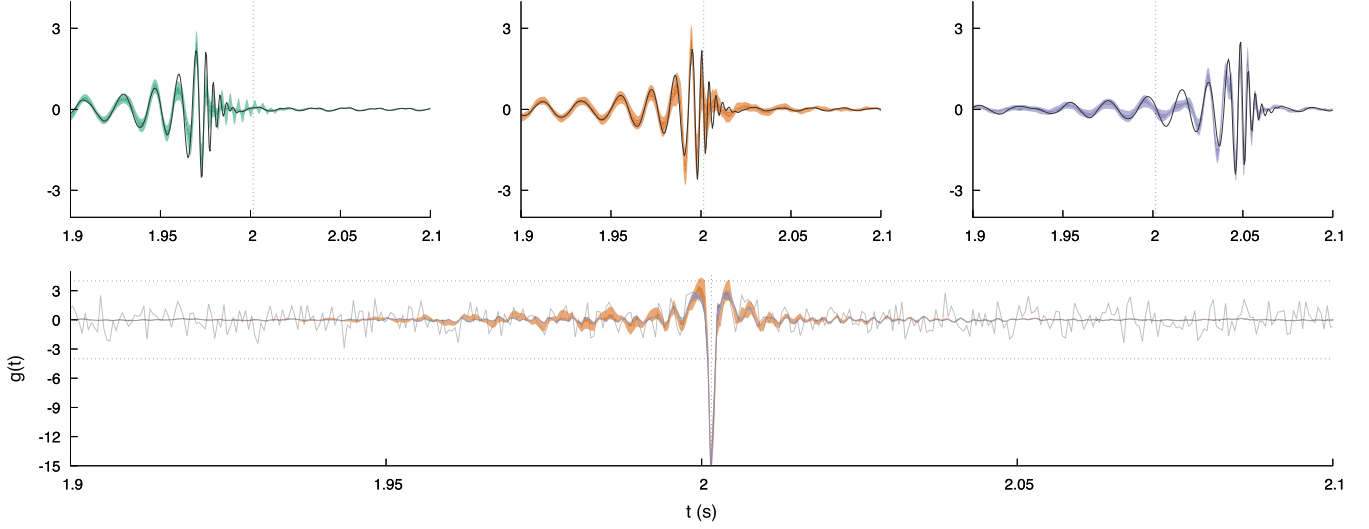


FIG. 3. Demonstration of the joint S + G analysis of data containing a noise transient in one detector and a coherent BBH signal in both data streams. The top panels show credible intervals for the whitened waveform reconstructions (colored) the true waveform (black) for three different injections with merger time just before, coincident with, and after the glitch, indicated by the vertical dashed line. The bottom panel shows the whitened data (gray) and the glitch reconstructions from each injection using the same color scheme as the top panel. The horizontal dashed lines indicate the vertical scale plotted in the top panel. The joint model accurately separates the GW signal from the glitch despite the degree to which the two features in the data overlap in time and frequency. The test was performed on O1 data containing a common glitch type in one detector and otherwise clean in the other. The simulated GW events had parameters similar to GW150914 and were added to the data before analysis with BayesWave.

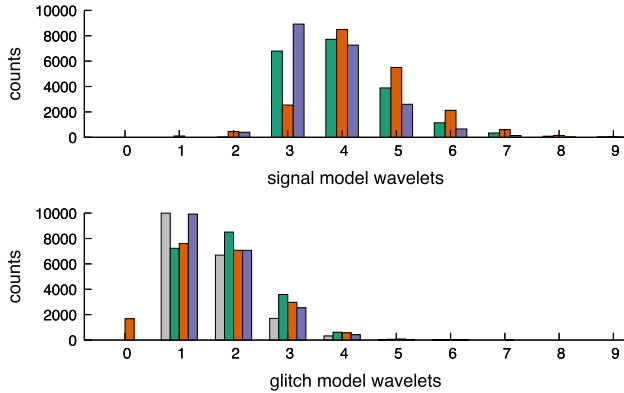


FIG. 4. Histograms of the total number of signal model wavelets and glitch model wavelets for the signal + glitch model analyses shown in Fig. 3. The color coding (green, orange, purple) corresponds to the three signal injections shown in the upper panel of Fig. 3. The additional gray colored bars are for when no signal is injected.

prior on the signal-to-noise ratio of an individual wavelet. For Morlet-Gabor wavelets the signal-to-noise ratio is estimated as

$$\rho^2 \simeq \frac{A^2 Q}{2\sqrt{2\pi} f_0 S_n(f_0)}, \quad (12)$$

where A , Q , f_0 are the time, quality factor, and central frequency of the wavelet respectively, and $S_n(f_0)$ is the one-sided noise power spectral density at f_0 .

Because for both astrophysical signals and instrumental glitches we expect to get many more low ρ events than loud events, we formulate priors that peak at a given ρ_* , and drop off at large and small ρ . Having the prior go to zero at low ρ helps with convergence: low amplitude wavelets have little effect on the likelihood and are disfavored by the natural parsimony of Bayesian inference. They eventually get discarded from the model, but it can take many iterations to shake them off. By shaping the prior to additionally disfavor low amplitude wavelets the convergence is accelerated. These convergence considerations apply equally to the signal and glitch models, and consequently, the same ρ_* value is used for both models. Values between $\rho_* = 3$ and $\rho_* = 10$ have been found to yield similar results. The default value is set at $\rho_* = 5$. For the glitch model the prior is

$$p(\rho|\text{glitch}) = \frac{\rho}{2\rho_*^2(1 + \frac{\rho}{2\rho_*})^3}. \quad (13)$$

For the signal model, the prior is

$$p(\rho|\text{signal}) = \frac{3\rho}{4\rho_*^2(1 + \frac{\rho}{4\rho_*})^5}. \quad (14)$$

The signal prior drops off as ρ^{-4} , as expected from the distribution astrophysical sources. The glitch prior has been designed to have a heavier tail at large ρ , since we expect to have more loud instrumental glitches than loud

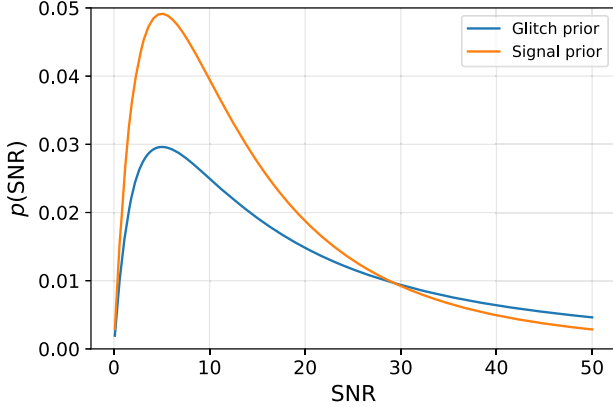


FIG. 5. Prior probability distribution on the signal-to-noise of individual wavelets, for both the signal and glitch model. In this example both priors peak at $\rho_* = 5$.

astrophysical signals. The priors on ρ for the signal and glitch models are shown in Fig. 5.

2. Wavelet dimension prior

In the original BayesWave configuration, the prior on the number of wavelets (N_w) was flat. After LIGO's first observing run, the distribution of wavelets in real data was used to develop a new prior. Using an analytic fit to the 500 most significant background triggers for the BayesWave's unmodeled transient search of O1 data [37], the wavelet distribution was empirically modeled as

$$p(N_w) = \frac{4\sqrt{3}N_w}{2\pi b^2(3 + \frac{N_w}{b})^4}, \quad (15)$$

where $b = 2.9$.

3. Sky position

For the extrinsic parameters, a feature was introduced to fix the sky location of the GW source to a known value. This is used for cases where there is a known electromagnetic counterpart to the GW signal, such as the kilonova associated with GW170817. Using a fixed sky location speeds up run times, and can improve waveform reconstructions.

IV. SAMPLER UPDATES

In addition to changes made to the underlying data models tested by BayesWave, there has been a similar scale of development to improve the efficiency of the pipeline. Improved performance can be found in two obvious places for an MCMC algorithm like BayesWave—shortening the burn-in time when the sampler is locating the most likely modes in the distribution, and increasing the sampling efficiency, thereby decreasing the autocorrelation length, of the chain samples.

Satisfying both goals at once, we have continued developing customized proposal distributions used by the MCMC sampler. In general, proposal distributions which are good approximations to the target distribution will locate the important modes of the posterior more rapidly, and sample the distribution with a smaller auto-correlation length. Below we describe proposals designed to leverage domain knowledge acquired either from a theoretical basis of how the likelihood function depends on the parameters, or proposals built from the input data itself.

In service of improving convergence time, we also describe a new model initialization procedure to get the sampler in a good starting position for the many components of the model, particularly in the case where large amplitude glitches bias the initial estimate of the noise spectrum which would then require a large number of samples to reduce down to the most parsimonious fit to the data.

A. Sky location proposal

The sky location and source orientation are poorly constrained for a two-detector network, often exhibiting multiple posterior modes that are a challenge to sample. These degeneracies are reduced with the addition of data from additional detectors, though the improvement can be small if the additional detectors are less sensitive than the original pair. In an effort to improve convergence of the sampler, we have introduced dedicated proposals to update the right ascension and declination of the source, (α, δ) , and the amplitude A , initial phase ϕ_0 and orientation of the source, described by the polarization angle ψ and ellipticity ϵ . While the sky proposal we use is only strictly valid for a two detector network and for sources with fixed elliptical polarization, it can be used for multidetector networks, though the acceptance rate will be reduced in those cases.

The sky-ring proposal selects a new sky location (α_y, δ_y) such that the time delay between the detectors is preserved. This requires an overall time shift dt to be applied to the waveform, which can be computed from the difference in the time of arrival at each detector for the current (α_x, δ_x) and proposed (α_y, δ_y) location. The sky ring is defined by rotations about the vector $\hat{z}$ connecting the vertices of the two detectors. Defining the angle between the current signal propagation vector $\hat{k}_x$ and $\hat{z}$: $\theta = \text{acos}(\hat{z} \cdot \hat{k}_x)$, we can construct the orthonormal triad $(\hat{u}, \hat{v}, \hat{z})$ with

$$\begin{aligned} \hat{u} &= \frac{\hat{k}_x - \cos \theta \hat{z}}{\sin \theta} \\ \hat{v} &= \frac{\hat{z} \times \hat{k}_x}{\sin \theta}. \end{aligned} \quad (16)$$

The new sky location is then found by drawing an angle ϕ uniformly in $[0, 2\pi]$ and rotating about $\hat{z}$ to yield

$$\hat{k}_y = \sin \theta (\cos \phi \hat{u} + \sin \phi \hat{v}) + \cos \theta \hat{z}. \quad (17)$$

The proposal density for the sky-ring move is constant, and cancels in the Metropolis-Hastings ratio. Since BayesWave references the time of arrival to a reference detector (usually Hanford), there is no need to shift the arrival time. If, however, the time of arrival is referenced to the Geocenter, then the arrival time needs to be shifted by an amount $\Delta t = \mathbf{R}_i \cdot (\hat{k}_y - \hat{k}_x)$, where $\mathbf{R}_i$ is the position of detector i relative to the Geocenter. If there are three or more detectors the sky-ring proposal can be used by picking a pair of detectors i, j . The acceptance of the sky-ring proposal will be lower in multidetector networks since the mapping only keeps the arrival time constant for that one pair of detectors.

The sky-ring proposal on its own has a low acceptance rate even for a two detector network. This is because the projection of the signal onto the detectors depends on the sky location. For a two detector network, and for signals with fixed elliptical polarization, we can find new extrinsic parameters, polarization angle ψ , ellipticity ϵ , amplitude A and overall phase ϕ_0 such that the waveforms in each detector are identical at the new sky location. This mapping greatly improves the acceptance rate of the sky-ring proposal.

The requirement that the waveform projections are the same at the current and proposed sky location yields the set of four equations:

$$\begin{aligned} u_x f_{1+x} + v_x f_{1 \times x} &= u_y f_{1+y} + v_y f_{1 \times y} \\ w_x f_{1+x} + z_x f_{1 \times x} &= w_y f_{1+y} + z_y f_{1 \times y} \\ u_x f_{2+x} + v_x f_{2 \times x} &= u_y f_{2+y} + v_y f_{2 \times y} \\ w_x f_{2+x} + z_x f_{2 \times x} &= w_y f_{2+y} + z_y f_{2 \times y}, \end{aligned} \quad (18)$$

where the f_{i+} and $f_{i \times}$ are the primitive antenna patterns in the i th detector, which are related to the full antenna patterns by

$$\begin{aligned} F_+(\alpha, \delta, \psi) &= f_+(\alpha, \delta) \cos(2\psi) + f_\times(\alpha, \delta) \sin(2\psi) \\ F_\times(\alpha, \delta, \psi) &= -f_+(\alpha, \delta) \sin(2\psi) + f_\times(\alpha, \delta) \cos(2\psi). \end{aligned} \quad (19)$$

The quantities u, v, w, z are defined such that

$$\begin{aligned} u &= A(\cos \phi \cos(2\psi) + \epsilon \sin \phi \sin(2\psi)) \\ v &= A(\cos \phi \sin(2\psi) - \epsilon \sin \phi \cos(2\psi)) \\ w &= A(\sin \phi \cos(2\psi) - \epsilon \cos \phi \sin(2\psi)) \\ z &= A(\sin \phi \sin(2\psi) + \epsilon \cos \phi \cos(2\psi)). \end{aligned} \quad (20)$$

We can solve for $\{u_y, v_y, w_y, z_y\}$:

$$\begin{aligned} u_y &= \frac{v_x f[\times_y \times_x \times_x \times_y] + u_x f[\times_y +_x +_x \times_y]}{f[\times_y +_y +_y \times_y]} \\ v_y &= \frac{v_x f[\times_x +_y +_y \times_x] + u_x f[+_x +_y +_y +_x]}{f[\times_y +_y +_y \times_y]} \\ w_y &= \frac{z_x f[\times_y \times_x \times_x \times_y] + w_x f[\times_y +_x +_x \times_y]}{f[\times_y +_y +_y \times_y]} \\ z_y &= \frac{z_x f[\times_x +_y +_y \times_x] + w_x f[+_x +_y +_y +_x]}{f[\times_y +_y +_y \times_y]}, \end{aligned} \quad (21)$$

where $f[abcd] = f_{1a}f_{2b} - f_{1c}f_{2d}$. Our next task is to invert the expressions for (u, v, w, z) to solve for $(A, \psi, \epsilon, \phi)$. Some algebra yields

$$\phi_y = \frac{1}{2} \text{atan}(q), \quad (22)$$

where

$$q = \frac{2(u_y w_y + v_y z_y)}{(w_y^2 + z_y^2) - (u_y^2 + v_y^2)}. \quad (23)$$

For $0 \leq \phi_y \leq \pi/4$, and $3\pi/4 \leq \phi_y \leq \pi$ the ellipticity is given by

$$\begin{aligned} \epsilon_y^{-1} &= \frac{u_y^2 + v_y^2 + w_y^2 + z_y^2}{2(u_y z_y - v_y w_y) \sqrt{1 + q^2}} + \frac{(w_y^2 + z_y^2) + 2q(u_y w_y + v_y z_y)}{2(u_y z_y - v_y w_y) \sqrt{1 + q^2}} \end{aligned} \quad (24)$$

and the amplitude is given by

$$\begin{aligned} A_y &= \left(\frac{(\sqrt{1 + q^2} + 1)(u_y^2 + v_y^2) + 2q(u_y w_y + v_y z_y)}{2\sqrt{1 + q^2}} + \frac{(\sqrt{1 + q^2} - 1)(w_y^2 + z_y^2)}{2\sqrt{1 + q^2}} \right)^{1/2}. \end{aligned} \quad (25)$$

For $\pi/4 < \phi_y < 3\pi/4$ the ellipticity is given by

$$\begin{aligned} \epsilon_y &= \frac{(u_y^2 + v_y^2 + w_y^2 + z_y^2) \sqrt{1 + q^2} + (u_y^2 + v_y^2)}{2(u_y z_y - v_y w_y) \sqrt{1 + q^2}} - \frac{(w_y^2 + z_y^2) + 2q(u_y w_y + v_y z_y)}{2(u_y z_y - v_y w_y) \sqrt{1 + q^2}}, \end{aligned} \quad (26)$$

and the amplitude is given by

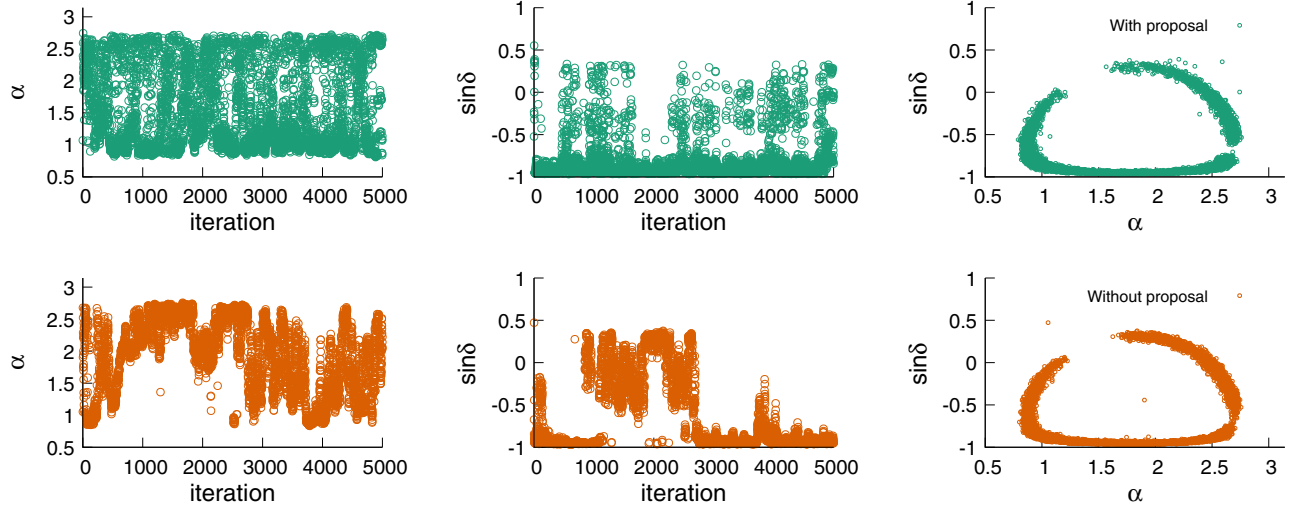


FIG. 6. Scatter plot of chain samples for sky localization parameters with (upper panel, green) and without (lower panel, orange) using the sky-ring proposal. Shown here is a subset of a full chain to highlight the difference in mixing between the settings. The panels from left to right show the right ascension (α) and sine of the declination ($\sin \delta$) as a function of chain iteration, and then the combined scatter plot of all chain samples. We use the data containing GW150914 for this demonstration.

$$A_y = \left(\frac{(\sqrt{1+q^2} + 1)(u_y^2 + v_y^2) + 2q(u_y w_y + v_y z_y)}{2\epsilon_y^2 \sqrt{1+q^2}} + \frac{(\sqrt{1+q^2} - 1)(w_y^2 + z_y^2)}{2\epsilon_y^2 \sqrt{1+q^2}} \right)^{1/2}. \quad (27)$$

Defining

$$\begin{aligned} g &= \left(1 + \sqrt{1+q^2}\right) v_y + q z_y \\ h &= \left(1 + \sqrt{1+q^2}\right) u_y + q w_y \end{aligned} \quad (28)$$

the polarization angle is given by

$$\psi_y = \begin{cases} \frac{1}{2} \text{atan}_2(g, h) & \text{for } 0 \leq \phi_y \leq \pi/4 \\ \frac{1}{2} \text{atan}_2(\epsilon_y h, \epsilon_y g) & \text{for } \pi/4 < \phi_y \leq 3\pi/4 \\ \frac{1}{2} \text{atan}_2(-g, -h) & \text{for } 3\pi/4 < \phi_y \leq \pi. \end{cases} \quad (29)$$

Using the above formulas, the phase ϕ_y covers the range $[0, \pi]$. To cover the full range $\phi_y \in [0, 2\pi]$ we also need to include the solutions found by setting $\phi_y \rightarrow \phi_y + \pi$ and $\psi_y \rightarrow \psi_y + \pi/2$, while keeping A_y and ϵ_y fixed.

The deterministic mapping $\vec{x} \rightarrow \vec{y}$ given above requires a nontrivial Jacobian in the Metropolis-Hastings ratio given by $J = |\partial \vec{y} / \partial \vec{x}|$. Numerical central differences are then used to compute $\partial y^i / \partial x^j$.

Figure 6 shows how the sky-ring proposal improves the mixing of the Markov chain and therefore the convergence

time of the analysis using GW150914 as an example. The figure panels show the trace of the sky location parameters' ($\alpha, \sin \delta$) samples as the chain iterates. Without the sky-ring proposal the sampler would have to run for longer to infer the correct relative weights between the different high-probability regions of the sky.

B. TFQ proposal

For trans-dimensional MCMC algorithms to efficiently sample in dimension space, well-designed proposal distributions which leverage domain knowledge are important. In addition to what is described in detail in [2], we have developed a new proposal to determine where wavelets should be placed in parameter space.

The proposal density is proportional to the matched filter signal-to-noise ratio ρ maximized over the wavelet phase, computed on a grid in time-frequency space with resolution of 5 ms in time and 4 Hz in frequency. The Q-scan is repeated for several different “layers” in Q , which has units of time, using a grid spacing of 2 s. The result is a three-dimensional discretized grid in time-frequency- Q space proportional to ρ . The distribution is then normalized by $(\sum_{ijk} \rho_{ijk}^2)^{1/2}$ where i, j, k are indices denoting the t, f , and Q grid location. The distribution is used to propose new wavelets in the fit, or to update current wavelet locations by rejection sampling uniform draws from $\{t, f, Q\}$ (TFQ) volume. The remaining wavelet parameters ($\mathcal{A}$ and ϕ_0) are drawn from the prior.

Figure 7 shows two-dimensional slices of the TFQ proposal at different Q layers increasing from top left to bottom right. Note how, as Q changes, so too does the aspect ratio of features highlighted in the time-frequency map. Different signal morphologies will be better

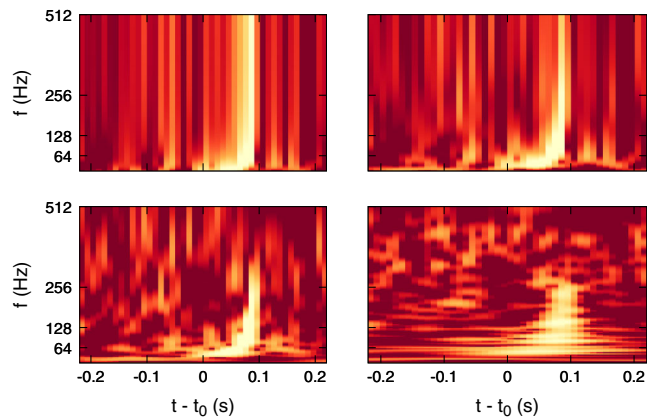


FIG. 7. Two-dimensional slices in the time-frequency plane of the TFQ proposal at increasing Q layers from top left to bottom right. The input data contains GW150914. The proposal will preferentially select draws from the Q layers where the signal appears most compactly, thereby maximizing the ρ per cell in the grid.

represented by different Q layers, which in turn will provide better sampling efficiency. The example data used in this figure contains the BBH event GW150914. The characteristic “chirp” shape is most clearly and compactly represented at the middle Q layers shown here (top right and bottom left) and would therefore be preferentially chosen for proposing updates to the wavelet model.

The TFQ proposal is most impactful as a proposal for transdimensional moves, when adding or removing a wavelet from the fit, but is also part of the proposal cycle for within dimension moves, taking an existing wavelet and proposing to replace it with a fair draw from the proposal.

C. Fast start PSD and glitch model

The BayesWave noise model may use many hundreds of parameters to describe the spline control points, the Lorentzian lines and the glitch model wavelets. Consequently, it can take hundreds of thousands of iterations for the sampler to reach the equilibrium distribution (“burn-in”), especially when loud glitches are present. The burn-in time can be reduced significantly by starting the chains at a good initial solution for the power spectral density and glitch model. We have adopted the fast deterministic method for iteratively estimating the power spectral density (PSD) and finding a maximum likelihood solution for the glitch model that is part of the low latency GlitchBuster algorithm, which can be used to remove noise transients from the LIGO/Virgo data in real time.

The GlitchBuster algorithm works as follows: The first step is to find a robust estimate for the power spectral density. Estimating the PSD for stationary, Gaussian noise is straightforward, and there are many methods to choose from. It is much more challenging to estimate the PSD for data that has a combination of Gaussian noise,

nonstationary noise transients (glitches), and long duration nonstationarity that causes the PSD estimate to vary with time. GlitchBuster employs an iterative approach using the same short data segments analyzed by BayesWave. The first step is to apply a Tukey window then FFT the data to compute the power spectrum. A running median is used to smooth the spectrum. Choosing the width of the smoothing window involves a trade-off between spectral distortion and smoothness. If the window is too wide the smoothed spectrum underestimates the slope of the power spectrum, while if the window is too short the resulting spectrum will not be very smooth. We also want the window to be wide enough so that sharp spectral line features get flagged as outliers. For the short 4–8 second data segments that are typically analyzed by BayesWave, we employ a 16 Hz window across much of the band, with a smaller 8 Hz window below 64 Hz, and a 4 Hz window below 32 Hz where the spectrum is very steep. With longer data segments the windows can be smaller since there are more frequency samples per Hz. Line features are identified as regions where the raw PSD exceeds the running median by some factor, typically set at 10. The full PSD model is the sum of the running median and the outliers. This initial PSD estimate can be biased by glitches, so the next step is to identify and remove the glitches. To do this the data is first whitened using the initial estimate for the PSD, then wavelet transformed using an overcomplete basis of continuous Morlet-Gabor wavelets. Wavelet denoising [38] is then used to remove regions of excess power (the glitches). The denoised data is then returned to the frequency domain, and the PSD estimation procedure is repeated. The original data is whitened using the new PSD estimate, followed by another round of wavelet denoising. This cycle is repeated until the ρ of the glitch model stabilizes, which typically takes between one and five iterations. The entire procedure takes approximately one second for a four second data segment, taking proportionally longer as the segment length increases.

The PSD estimate, made up of a smooth component and a collection of outliers, is next mapped to the parameters used by the BayesLine Bayesian spectral estimation algorithm. The smooth component of the PSD is used to compute the initial cubic spline model using a fixed frequency spacing, typically set equal to the minimum spacing allowed by the BayesLine spline model. The outliers are mapped onto the Lorentzian line model by finding the central frequency, frequency extent and maximum height of each outlier region, and using these to compute the central frequency, amplitude and scale of the Lorentzian function that approximates the outliers.

With the initial PSD model in hand, the next step is to solve for the glitch model in a form that can be used by BayesWave. While the wavelet denoising procedure used in the PSD estimation produces a glitch model, it is not in a form that can be used by BayesWave. Moreover, the denoising is performed using wavelets with a single quality factor, and much better fits can be found using wavelets

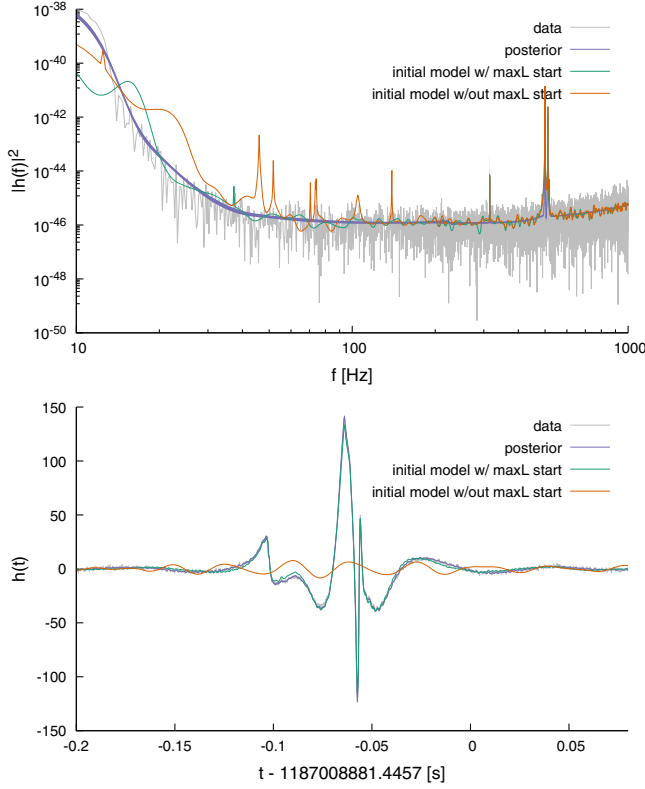


FIG. 8. Demonstration of the improved model initialization from GlitchBuster using the glitch in the Livingston data near GW170817 [16]. The top panel shows the data (gray), and PSD estimates with (green) and without (orange) the maximum likelihood wavelet initialization step from GlitchBuster. These are compared to the posterior after the sampler has finished (purple). The excess power in the PSD model shown in the orange curve is due to the BayesLine parameters fitting part of the glitch. The bottom panel shows the whitened glitch model compared to the data. The orange curve is from the initialization of the wavelet model that uses a fair draw from the prior. Note that the GlitchBuster wavelet model starts at a solution where the reconstruction is consistent with the posterior. The median of the PSD posterior was used for whitening each of the glitch reconstructions in the bottom panel.

with a range of quality factors. To that end, the whitened data is wavelet transformed at geometric sequence quality factors using continuous Morlet-Gabor wavelets on a grid in time and frequency, following the identical procedure used to produce the TFQ proposal. The loudest pixel in the TFQ map is found, and if it exceeds some ρ threshold (e.g., $\rho = 3$), the corresponding wavelet is subtracted from the data. Since continuous wavelets overlap with their neighbors, the TFQ map has to be updated in a region surrounding the wavelet that was removed. The procedure is repeated with the updated TFQ map until no significant outliers remain. The parameters of the loud wavelets identified in this way can now be used as a starting point for the BayesWave glitch model. The iterative subtraction procedure is improved by adding a likelihood maximization

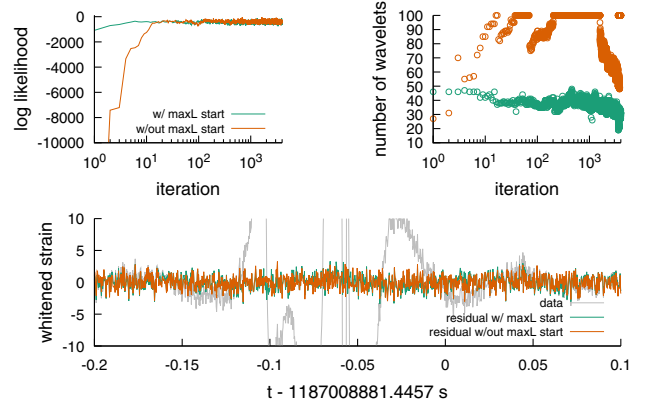


FIG. 9. Demonstration of convergence time improvement when using the PSD and glitch model initialization on a high ρ glitch in the Livingston detector during the binary neutron star merger GW170817. The top left panel shows the log likelihood chain of the sampler, top right shows the number of wavelets used in the fit, and the bottom panel shows the residual after glitch subtraction. The orange curves are for the version of BayesWave using a random starting point for the chains. The green traces are for the sampler that uses the new initialization step. Both versions achieve a similar fit to the glitch (as shown in the likelihood and residual plots) but the green curves achieve that fit with a smaller number of wavelets, which the orange curve was slowly trending towards before the sampler was stopped.

step after each subtraction, similar to the F-statistic procedure [39], but using the collection of wavelets as the filter functions. Figure 8 shows an example of the maximum likelihood PSD initialization (upper panel) and glitch model initialization (lower panel).

Figure 9 compares the convergence on data containing a high amplitude noise transient with (green) and without (orange) using the GlitchBuster initialization. The top left panel shows the likelihood chain for the sampler, both of which reach similar values although the GlitchBuster-initialized chain achieves that value $\sim \mathcal{O}(10)$ times faster. However, the number of wavelets used by the naive start (top right panel) is larger, and tens of thousands of sampler iterations were required for the model to even begin sampling with the more parsimonious number of wavelets. The bottom panel shows that in both cases the glitch is adequately removed from the data (gray) by comparing the residuals (orange and green), also indicated by the comparable likelihoods achieved by both chains. The data used in this example contains the glitch just before the GW170817 merger [16].

V. POSTPROCESSING

While the output of the RJMCMC in BayesWave is samples from the posterior distribution of wavelet parameters and extrinsic parameters common to the signal (such as sky location, polarization information), ultimately it is the morphology and properties of the GW signal that are of interest. Below we discuss the postprocessing used to translate raw samples into more meaningful outputs.

A. Waveform reconstructions

For each sample in the resulting posterior, the wavelet parameters are summed to produce a GW waveform $\mathbf{h} = h(t_i)$, where $i \in \{1, N_t\}$ with N_t the number of discrete time steps.¹ For each time step t_i , the BayesWave postprocessing calculates the median and bounds of the 50% and 90% credible intervals on the posterior distribution of $h(t_i)$. An example of a waveform reconstructions is shown in Fig. 10.

BayesWave also produces a posterior distribution on $\tilde{h}(f)$, the GW waveform in the frequency domain. The median and credible intervals for $\tilde{h}(f)$ are calculated in the same manner as described above.

B. Frequency evolution reconstructions

In addition to the time and frequency domain waveforms, one may also be interested in looking at the frequency evolution over time of the GW signal (for example, one may wish to look for the characteristic chirp of a compact binary inspiral). The BayesWave postprocessing produces posterior distributions of this frequency evolution, $f(t)$. This $f(t)$ is found by using points where $h(t) = 0$ (also called the zero crossings). The frequency at the j th zero crossing, denoted t_j^0 is given by

$$f(t_j^0) = \frac{1}{t_{j+1}^0 - t_{j-1}^0}. \quad (30)$$

The BayesWave postprocessing calculates $f(t)$ for each $h(t)$ posterior sample, and again calculates the median and 50% and 90% credible intervals. An example is shown in Fig. 10.

C. Waveform moments

Another way to characterize the signal is to calculate the central moments of the reconstructed waveform. BayesWave postprocessing calculates the first two central moments both in the time domain (central time and duration) and the frequency domain (central frequency and bandwidth) along with their probability distributions. In principle, we can calculate higher order moments too, but those are less intuitive, and expected to be measured less precisely. A comprehensive study of the parameter estimation capabilities of BayesWave (including estimation of waveform moments) has been presented in [12].

D. Whitening tests

When the PSD is estimated from on-source data using the BayesLine algorithm, a number of tests are performed in order to ensure that the computed PSD whitens the data. The tests are described in more detail in [9] and include histograms of the real and imaginary Fourier domain

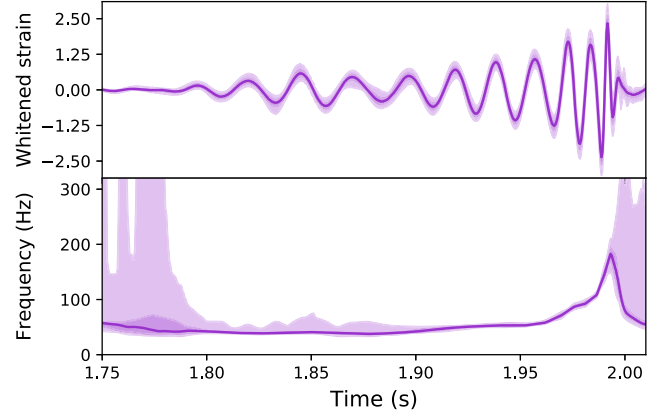


FIG. 10. Example of the reconstructions from the postprocessing phase for the whitened time domain waveform (top), and the frequency evolution over time (bottom) for the example event GW170814 [40]. The solid line is the median reconstruction, and the shaded bands are the 50% and 90% credible intervals. For the frequency evolution over time, the process described in Sec. VB uses the zero crossings of the reconstructed waveform, and as such times where the reconstructed strain is consistent with zero the $f(t)$ reconstruction will have large credible intervals in this region. This is seen in times before about 1.8 seconds, and after merger (about 2.0 seconds) in the lower plot.

residuals as compared to a $\mathcal{N}(0, 1)$ distribution, properties of the combined 2D distribution, and the Anderson-Darling test. The latter quantifies the degrees to which samples are drawn from a target distribution and results in a p-value for the null hypothesis that the Fourier residuals are draws from $\mathcal{N}(0, 1)$. The Anderson-Darling test is performed for various bandwidths, to ensure that the resulting PSD whitens all relevant frequency ranges of the analysis. An example of a histogram of the whitened Fourier residuals is shown in Fig. 11 corresponding to 4s of LIGO-Hanford data around GW150914. The p-value for the plotted data is 0.81, suggesting that the null hypothesis that the residuals are drawn from $\mathcal{N}(0, 1)$ cannot be ruled out.

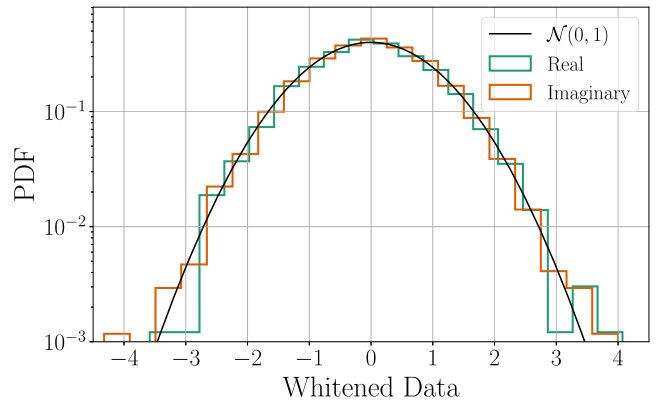


FIG. 11. Histogram of the real and the imaginary part of the whitened Fourier residuals of 4 s of data around GW150914. The noise power spectral density has been computed with BayesWave and the target $\mathcal{N}(0, 1)$ distribution is shown for reference.

¹The number of discrete time steps is determined by the sampling rate and segment length used in the BayesWave analysis.

VI. REVIEW TESTS

BayesWave and BayesLine periodically undergo standard review tests for sampling algorithms. Perhaps the most common (and computationally inexpensive) review test is the “constant likelihood test,” i.e., run the code in a configuration where the likelihood function is a constant. In this case the posterior is equal prior and the resulting samples must follow the prior distributions for all model parameters. This test ensures that sampling satisfies detailed balance and it does indeed produce fair samples from the posterior distribution. We routinely confirm that BayesWave passes this test.

Another common test (though computationally more expensive) that checks the likelihood itself is the “P-P test,” where simulated signals are drawn from the prior distribution of each parameter and then analyzed. If a sampler is unbiased, then the true value for each parameter must be at the p -th percentile of its posterior distribution for p events. Stated differently, a plot of the fraction of events where the true value is at a certain percentile of the posterior must be diagonal for each parameter. This test is stronger but more computationally intensive, requiring dozens of injections.

Figure 12 shows the results of the P-P test for the sky location (top panel) and the individual model parameters (wavelet parameters and extrinsic parameters) (bottom panel). Shaded regions denote 1-, 2-, 3- σ errors. In all cases we recover diagonal lines within the expected error.

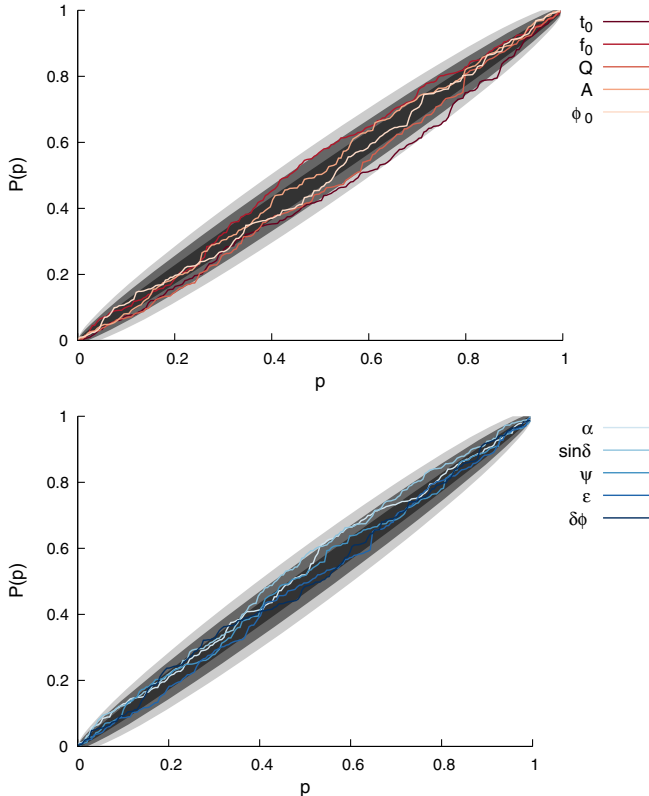


FIG. 12. P-P plots for model parameters and sky localization.

The study was performed using simulated signals and simulated noise.

VII. CONCLUSIONS

The BayesWave algorithm continues to be improved, with additional functionality added and efficiency achieved since the initial release. The most significant changes in the release described here include the ability to model signals with general polarization content, simultaneous modeling of signals and noise transients, and significantly improved sampling. With this added functionality the use cases for BayesWave continue to grow, including glitch subtraction, tests of general relativity, and independent checks of the waveform models used to describe compact binary mergers.

Work continues to further extend and improve the performance. Future updates will allow for the joint sampling of compact binary coalescence templates and noise transients in addition to marginalization over the power spectral density of the noise. This functionality will be particularly valuable for low mass systems, such as binary neutron star mergers, where the long duration of the signals makes it highly likely that noise transients will also be present in the data. New tests of general relativity will be made possible by extending the polarization model to allow for scalar and vector polarization states. Dynamic spectral modeling, where the power spectral density can change with time, will be made possible by switching the analysis from the Fourier domain to the discrete wavelet domain [41].

Looking further afield, the BayesWave approach is being applied to other branches of gravitational wave astronomy, including pulsar timing arrays [42,43] and the future space-based LISA detector [44].

ACKNOWLEDGMENTS

This research has made use of data, software and/or web tools obtained from the Gravitational Wave Open Science Center [45], a service of LIGO Laboratory, the LIGO Scientific Collaboration and the Virgo Collaboration. LIGO is funded by the U.S. 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. N.J.C. and B.B. acknowledge the support of NSF Grants No. PHY1607343 and No. PHY1912053. The authors are grateful for computational resources provided by the LIGO Laboratory and supported by National Science Foundation Grants No. PHY-0757058 and No. PHY-0823459, and for resources provided by the Open Science Grid [46,47], which is supported by the National Science Foundation Grant No. 1148698, and the U.S. Department of Energy’s Office of Science. The Flatiron Institute is supported by the Simons Foundation. The J.A.C. and S.G. gratefully acknowledge the NSF for financial support from

Grants No. PHY 1806580, No. PHY 1809572, and No. TG-PHY120016. Parts of this research were conducted by the Australian Research Council Centre of Excellence for Gravitational Wave Discovery (OzGrav), through Project No. CE170100004.

APPENDIX A: SOFTWARE AND WORKFLOW

BayesWave is an open source software project distributed under the terms and conditions of the Gnu Public License (GPL2). Source code and documentation can be found at [6]. Code development takes place via GitLab flow, with the core development team contributing and enhancing the software through regular merge requests from feature branches on their own forks of the repository.

The BayesWave software stack and workflow is comprised of four principal components: BayesWavePipe (written in PYTHON): a workflow generation tool; BayesWave (C): the main analysis application which reads and conditions data, and fits for the various models described in this work; BayesWavePost (C): postprocessing application which parses the results from BayesWave to produce reconstructed time- and frequency-domain representations of the models and cleaned data. Results are presented through a collection of PYTHON-based plotting and web-page generation scripts. Given a single time to analyze, along with a configuration file detailing the time-frequency volume of interest and the analysis mode, BayesWavePipe is used to construct a simple HTCondor [48] DAGMan workflow in which each of these applications is executed in serial. The process is trivially extended to larger analyses, such as Monte-Carlo simulations signal detection and characterization or background trials for significance estimation, by executing the same workflow as many times as required but as parallel, independent jobs on high-throughput computing resources.

BayesWave is packaged with conda [49] and technically and scientifically reviewed releases are available from the standard IGWN conda environment [50] which is distributed via the OASIS CVMFS [51] repository managed by the Open Science Grid project [52,53]. Docker images with the most recent release and the current state of the main git branch are available from the container registry associated with the source code repository. Finally, these docker images are distributed to the OSG CVMFS container repository (where they are automatically converted to singularity images), provisioning access to the latest and development versions of the code, in addition to the reviewed packages in conda, on OSG resources.

APPENDIX B: WORKFLOW CHARACTERIZATION

In this Appendix we characterize typical BayesWave and BayesLine analysis job profiles to provide an idea of the computational resources required. Exemplary scenarios

selected include power spectral density estimation for BBH-like events using BayesLine and gravitational wave signal model-only analysis, both in a variety of time-frequency configurations used for previous LIGO-Virgo detections. While there is some further variation when evaluating the glitch or signal-plus-glitch models, the signal-only model serves, to first order, to illustrate the resource consumption per model when running BayesWave. For example, the wall time to compute a signal-versus-glitch Bayes factor for one of the configurations shown here can be estimated from the BayesLine wall time, plus $2\times$ the BayesWave wall time.

Our figures of merit for each configuration are the cumulative wall clock and CPU times, the peak memory usage and the peak disk usage, recorded by HTCondor over the lifetime of the job. Workflows used in this characterization ran from Singularity containers, using native support in HTCondor, and images deployed in the OSG CVMFS repository. All calculations were performed on Intel Xeon E3-1240 v5 processors in the dedicated LIGO computing cluster at the California Institute of Technology.

Finally, it is worth noting that, due to its long run times (up to nearly two days in some cases considered here and often longer in more extreme cases), BayesWave saves the state of the calculation every hour and exits, to be resumed by the workflow management system, in order to avoid data loss in case of worker node contention or loss of connectivity. We find this process has a negligible detrimental impact on the time to solution for continuously running jobs, while being absolutely critical to operating on shared resources.

1. BBH PSD estimation

BayesLine power spectral density estimates have proven a critical component of upstream, template-based parameter estimation efforts. To characterize BayesLine performance we have reanalyzed 24 BBH events from the GWTC-2 catalog [18] using identical time-frequency configurations to those used in the LIGO-Virgo parameter estimation studies.² Each event was analyzed with ten independent trials to account for fluctuations in system load, and network/disk performance. Two sets of results are produced: one set using the “low-latency” BayesLine configuration, with 100000 MCMC iterations, which was used to inform online parameter estimation efforts in O3, and a “high-latency” BayesLine configuration with 4000000 MCMC iterations.

Figure 13 shows, from top to bottom: the wall time, peak memory usage and peak disk space used by the main BayesWave RJMCMC program, running in the low-latency

²The O3a catalog events were analyzed with a variety of time-frequency configurations: higher-mass BBH mergers result in shorter-duration, lower-frequency signals, necessitating analysis of a smaller time-frequency volume.

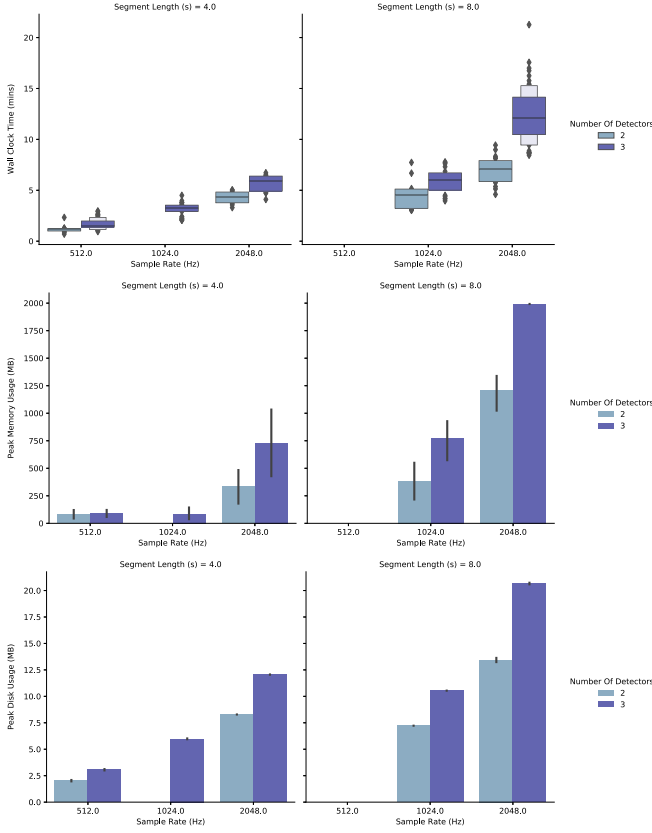


FIG. 13. Job characterization metrics for the main BayesWave RJMCMC process for low-latency BayesLine PSD estimation for LIGO-Virgo O3a catalog BBH events. Results are grouped by each event's time-frequency and detector configuration.

BayesLine configuration. CPU time is not measurably different from wall time for these jobs and is not shown here. Results from each of the 24 BBH events are grouped by the BayesLine time-frequency configuration and number of detector data streams used in the analysis. The median wall time for low-latency BayesLine PSD estimation ranges between ~ 1 minute, for the smallest time-frequency volumes considered here (two data streams with duration 4 s and sample frequency 512 Hz), to ~ 12 minutes for the three detector analyses with eight second segments, sampled at 2048 Hz. Note that the corresponding BayesWavePost jobs, which parse the sampled model parameters and reconstruct the inferred PSD, typically complete in under 30 seconds here, with minimal memory and disk footprints; their data is not shown.

Equivalent statistics for the off-line BayesLine PSD estimation are shown in Fig. 14. The time to solution has increased proportionately with the 40-fold increase in the number of MCMC iterations, with wall times now measured in hours: the PSD for the smallest time-frequency volume is now only attained after a median wall time of 40 minutes, while the longer eight-second segments require 7.5 hours.

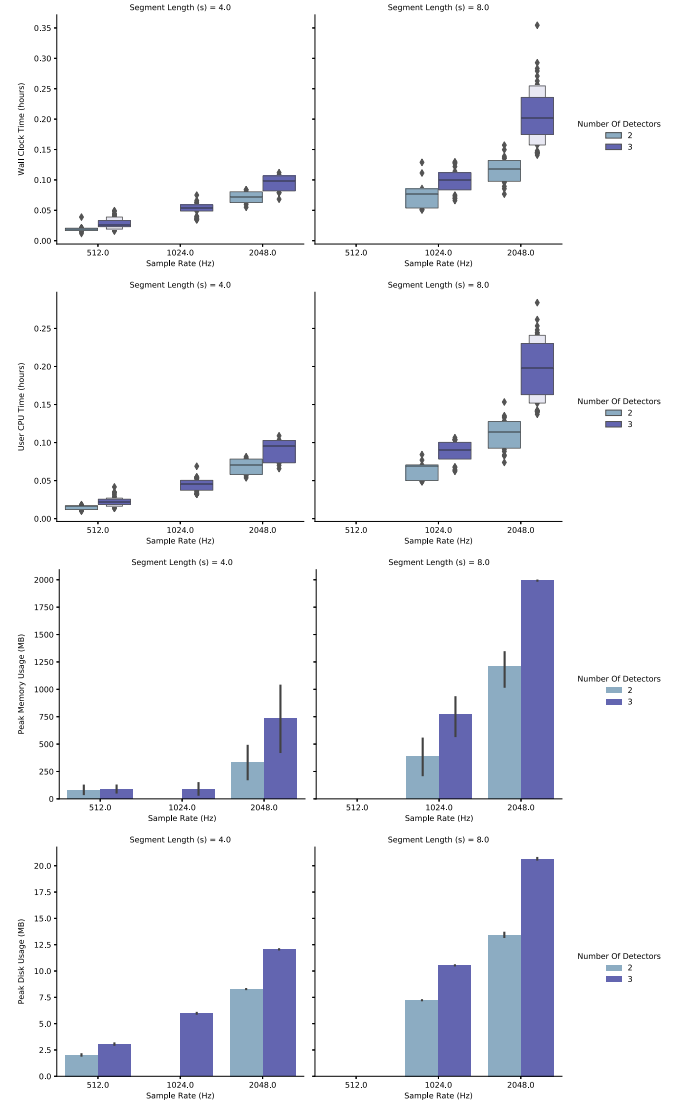


FIG. 14. Analysis job metrics for the main BayesWave RJMCMC process for final BayesLine PSD estimation for LIGO-Virgo O3a catalog BBH events.

Now that we are into a regime where jobs are running long enough as to require periodic checkpointing their progress, it is worth checking that the exit/resume behavior every hour is not detrimental to workflow efficiency. Figure 14 also shows the CPU time, as well as the wall time: The difference in the median wall and CPU time is less than 5% of the total wall time, while their distributions are, to all intents and purposes, identical. This provides some reassurance that there is no significant startup penalty incurred from saving and resuming jobs.

The difference in the memory footprint is more marginal, with that cost being dominated by storing an identical time-frequency maps of the data. Unsurprisingly, the storage requirements have also increased significantly, going from ~ 2 – 20 MB for the low-latency analysis, to ~ 18 – 130 MB for the off-line analysis.

As before, the BayesWavePost jobs do not add significantly to the total time to solution, taking, at most, about 15 minutes to complete and presenting nearly identical memory and disk footprints as the parent BayesWave jobs.

2. BBH waveform reconstructions

We now repeat the characterization above using the gravitational wave signal model. In these analyses, we compute the evidence for the signal model, sample the posterior probability distribution function for the signal model parameters to reconstruct the underlying gravitational wave signal present in the data stream from each detector, and we compute posterior probability density functions for a variety of moments of the reconstructed waveforms. Typically, the waveform reconstructions from such analyses are used to search the data for deviations from physically parametrized waveform models. As before, we reanalyze each of the BBH events from the LIGO-Virgo O3a catalog, using ten independent trials in each case to average over any fluctuations in e.g., system load or network performance. These waveform reconstruction analyses use the median off-line BayesLine PSDs

computed in the previous section. That is, the metrics obtained here are purely for the evaluation of the signal model and do not include any overhead for PSD estimation.

Finally, it should be noted that full BayesWave waveform reconstruction comparisons, like those in Sec. VIII of [18], typically use Monte-Carlo simulations of template-based waveform reconstructions, in addition to analyzing the actual gravitational wave signal. In assessing any expected total resource consumption based on the following measurements then, the reader should scale these results by the number of Monte-Carlo simulations to be performed [usually $\mathcal{O}(100)$].

Figure 15 show the wall time, peak memory consumption and the disk usage of the BBH events from the O3a catalog. Again, results are grouped by total time-frequency volume used in the analyses. As before, the difference in wall and CPU times is much smaller than the overall wall time so, in the interests of brevity, the latter is not shown. Evaluating the signal model now presents a more formidable computational challenge, with wall times in the range of 5–48 hours. Further, Fig. 16 shows the metrics for the corresponding BayesWavePost child jobs. While these jobs still typically complete in well under an hour, their memory requirements can be substantial, due to a need to store several thousand time-domain waveform reconstructions in

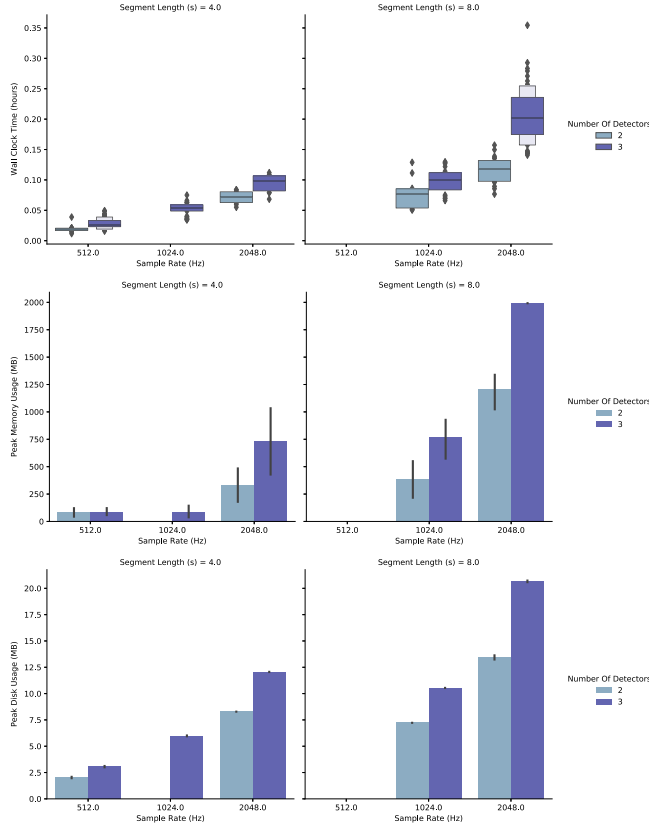


FIG. 15. Analysis job metrics for the main BayesWave RJMCMC process for evaluation of the gravitational wave signal model for the purposes of waveform reconstruction analyses. Configurations shown are those used for the LIGO-Virgo O3a catalog events.

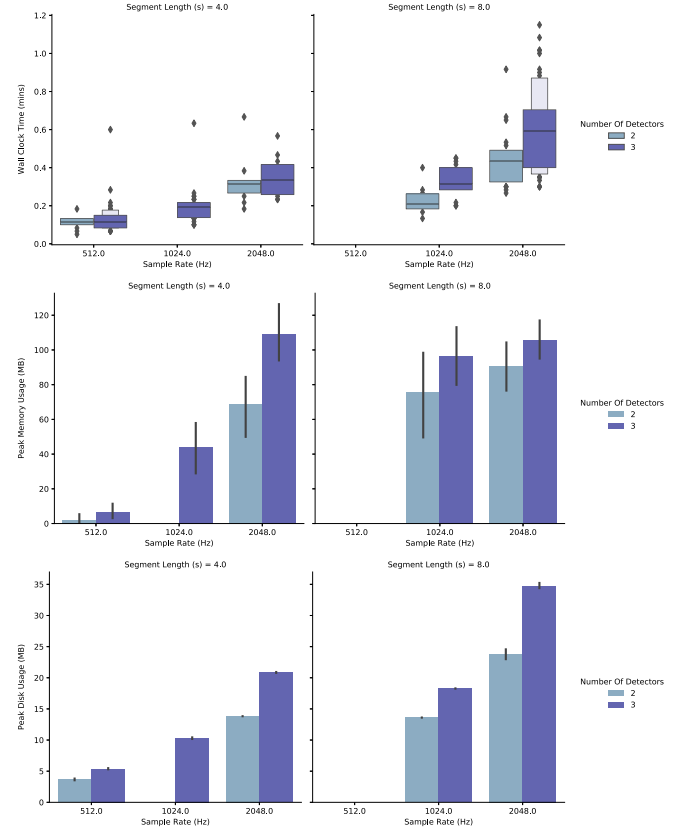


FIG. 16. Analysis job metrics for the BayesWavePost process, run subsequent to the jobs shown in Fig. 15.

memory—the median peak memory consumption gets as high as 11 GB for the 8 second segments considered here.

APPENDIX C: OPTIMIZATION

1. Delta likelihood updates

Computing the full likelihood (3) for a network with M detectors over a time span with N data points takes $\mathcal{O}(MN)$ operations. The computational cost of the likelihood calculation can be significantly reduced by performing “delta” updates that are localized in frequency. For example, the amplitude envelope of the Morlet-Gabor wavelets falls off as $\exp[-\pi^2\tau^2(f-f_0)^2]$, and to a good approximation, the likelihood only needs to be computed for frequencies $f_0 - \delta f < f < f_0 + \delta f$, with $\delta f \simeq 4/(\pi\tau) = 8f_0/Q$ when updating the contribution from a particular wavelet.

The delta likelihood updates work as follows. Suppose the model is currently given by h_x . The residual is $r_x = d - h_x$ and the log likelihood is given by $\ln L_x = -(r_x|r_x)/2 + W_x$, where W_x depends only on the noise model. Holding the noise model fixed and adding (removing) a single wavelet Ψ to the model yields the updated residual $r_y = r_x \mp \Psi$ and the updated likelihood

$$\ln L_y = \ln L_x + \delta \ln L, \quad (\text{C1})$$

where

$$\delta \ln L = (\Psi|\Psi)/2 \pm (\Psi|r_x). \quad (\text{C2})$$

The delta likelihood only has to be evaluated over frequency band $f_0 \pm \delta f$. The computational saving is significant for wavelets with large quality factors Q , and/or for wavelets with low central frequencies f_0 . The same delta likelihood method can be used to update the parameters of an existing wavelet in the model by first subtracting a wavelet with the current parameters, then adding a wavelet with the updated parameters.

2. Recursive evaluation of wavelets

The expressions (4) for the Morlet-Gabor wavelets contain real and complex exponentials that can be costly to evaluate. Replacing these function calls by recursion

relations significantly decreases the computational cost. The term $\exp[\pm 2\pi i(f-f_0)t_0 + \phi_0]$ in the frequency domain expression, and $\cos(2\pi f_0(t-t_0) + \phi_0)$ in the time domain expression, can be computed using trigonometric recursion relations. Picking a reference frequency f_* or reference time t_* , the phase terms can be written as $\Phi_n = \Phi_* + n\Delta\Phi$, where for the frequency domain case $\Phi_* = 2\pi(f_* - f_0)t_0 + \phi_0$ and $\Delta\Phi = 2\pi\Delta f t_0$, and for the time domain case $\Phi_* = 2\pi(t_* - t_0)f_0 + \phi_0$ and $\Delta\Phi = 2\pi\Delta t f_0$. The evaluation is initialized by computing the four terms $\cos \Phi_0$, $\sin \Phi_0$, $\cos \Delta\Phi$, $\sin \Delta\Phi$. Subsequent values are found by multiplication and addition according to the recursion relation:

$$\begin{aligned} \cos \Phi_{n+1} &= \cos \Phi_n \cos \Delta\Phi - \sin \Phi_n \sin \Delta\Phi \\ \sin \Phi_{n+1} &= \cos \Phi_n \sin \Delta\Phi + \sin \Phi_n \cos \Delta\Phi. \end{aligned} \quad (\text{C3})$$

The amplitude terms contain exponentials of the form $\exp[-Q^2 f/f_0]$, $\exp[-\pi^2\tau^2(f-f_0)^2]$, and $\exp[-(t-t_0)^2/\tau]$. The first of these terms is linear in the frequency increment Δf , while the later include quadratic terms in Δf or Δt . The term $A = \exp[-Q^2 f/f_0]$ can be computed using the recursion

$$A_{n+1} = A_n e^{-Q^2 \Delta f / f_0}, \quad (\text{C4})$$

with $A_0 = \exp[-Q^2 f_*/f_0]$. The terms with quadratic in the increments require a two-part recursion. For example, the frequency domain term $A = \exp[-\pi^2\tau^2(f-f_0)^2]$ can be computed using the recursion relation

$$\begin{aligned} A_{n+1} &= A_n \alpha_n e^{(2f_0\Delta f - \Delta f^2)\pi^2\tau^2} \\ \alpha_{n+1} &= \alpha_n e^{-2(\pi\tau\Delta f)^2} \end{aligned} \quad (\text{C5})$$

with

$$\begin{aligned} A_0 &\equiv e^{-(\pi^2\tau^2(f_*-f_0)^2)} \\ \alpha_0 &= e^{-2\pi^2\tau^2 f_* \Delta f}. \end{aligned} \quad (\text{C6})$$

The quadratic time-domain amplitude $\exp[-(t-t_0)^2/\tau]$ can be computed in a similar fashion.

-
- [1] B. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. Lett.* **116**, 061102 (2016).
 [2] N.J. Cornish and T.B. Littenberg, *Classical Quantum Gravity* **32**, 135012 (2015).

- [3] T.B. Littenberg and N.J. Cornish, *Phys. Rev. D* **91**, 084034 (2015).
 [4] J. Aasi *et al.* (LIGO Scientific Collaboration), *Classical Quantum Gravity* **32**, 074001 (2015).

- [5] F. Acernese *et al.* (Virgo Collaboration), *Classical Quantum Gravity* **32**, 024001 (2015).
- [6] BayesWave source code repository, <https://git.ligo.org/lscsoft/bayeswave/>.
- [7] J. D. Romano and N. J. Cornish, *Living Rev. Relativity* **20**, 2 (2017).
- [8] M. Millhouse, N. J. Cornish, and T. Littenberg, *Phys. Rev. D* **97**, 104057 (2018).
- [9] K. Chatziioannou, C.-J. Haster, T. B. Littenberg, W. M. Farr, S. Ghonge, M. Millhouse, J. A. Clark, and N. Cornish, *Phys. Rev. D* **100**, 104004 (2019).
- [10] A. Gelman and X.-L. Meng, *Stat. Sci.* **13**, 163 (1998).
- [11] T. B. Littenberg, J. B. Kanner, N. J. Cornish, and M. Millhouse, *Phys. Rev. D* **94**, 044050 (2016).
- [12] B. Bécsy, P. Raffai, N. J. Cornish, R. Essick, J. Kanner, E. Katsavounidis, T. B. Littenberg, M. Millhouse, and S. Vitale, *Astrophys. J.* **839**, 15 (2017).
- [13] J. B. Kanner, T. B. Littenberg, N. Cornish, M. Millhouse, E. Xhakaj, F. Salemi, M. Drago, G. Vedovato, and S. Klimenko, *Phys. Rev. D* **93**, 022002 (2016).
- [14] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. D* **93**, 122004 (2016); **94**, 069903(A) (2016).
- [15] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. D* **100**, 024017 (2019).
- [16] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. Lett.* **119**, 161101 (2017).
- [17] C. Pankow *et al.*, *Phys. Rev. D* **98**, 084016 (2018).
- [18] R. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), [arXiv:2010.14527](https://arxiv.org/abs/2010.14527).
- [19] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. X* **9**, 031040 (2019).
- [20] K. Chatziioannou, J. A. Clark, A. Bauswein, M. Millhouse, T. B. Littenberg, and N. Cornish, *Phys. Rev. D* **96**, 124035 (2017).
- [21] B. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. X* **9**, 011001 (2019).
- [22] A. Torres-Rivas, K. Chatziioannou, A. Bauswein, and J. A. Clark, *Phys. Rev. D* **99**, 044014 (2019).
- [23] G. Dálya, P. Raffai, and B. Bécsy, *Classical Quantum Gravity* **38**, 065002 (2021).
- [24] K. W. Tsang, M. Rollier, A. Ghosh, A. Samajdar, M. Agathos, K. Chatziioannou, V. Cardoso, G. Khanna, and C. Van Den Broeck, *Phys. Rev. D* **98**, 024023 (2018).
- [25] K. W. Tsang, A. Ghosh, A. Samajdar, K. Chatziioannou, S. Mastrogiiovanni, M. Agathos, and C. Van Den Broeck, *Phys. Rev. D* **101**, 064012 (2020).
- [26] K. Gill, W. Wang, O. Valdez, M. Szczepanczyk, M. Zanolin, and S. Mukherjee, [arXiv:1802.07255](https://arxiv.org/abs/1802.07255).
- [27] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. Lett.* **116**, 241102 (2016).
- [28] S. Ghonge, K. Chatziioannou, J. A. Clark, T. Littenberg, M. Millhouse, L. Cadonati, and N. Cornish, *Phys. Rev. D* **102**, 064056 (2020).
- [29] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. Lett.* **116**, 221101 (2016); **121**, 129902(E) (2018)].
- [30] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. D* **100**, 104036 (2019).
- [31] R. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), [arXiv:2010.14529](https://arxiv.org/abs/2010.14529).
- [32] M. Hannam, P. Schmidt, A. Bohé, L. Haegel, S. Husa, F. Ohme, G. Pratten, and M. Pürrer, *Phys. Rev. Lett.* **113**, 151101 (2014).
- [33] S. Fairhurst, R. Green, C. Hoy, M. Hannam, and A. Muir, *Phys. Rev. D* **102**, 024055 (2020).
- [34] G. Pratten, P. Schmidt, R. Busicchio, and L. M. Thomas, *Phys. Rev. Research* **2**, 043096 (2020).
- [35] P. Schmidt, F. Ohme, and M. Hannam, *Phys. Rev. D* **91**, 024043 (2015).
- [36] M. Cabero, A. Lundgren, A. H. Nitz, T. Dent, D. Barker, E. Goetz, J. S. Kissel, L. K. Nuttall, P. Schale, R. Schofield, and D. Davis, *Classical Quantum Gravity* **36**, 155010 (2019).
- [37] B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. D* **95**, 042003 (2017).
- [38] C. Torrence and G. P. Compo, *Bull. Am. Meteorol. Soc.* **79**, 61 (1998).
- [39] P. Jaranowski, A. Krolak, and B. F. Schutz, *Phys. Rev. D* **58**, 063001 (1998).
- [40] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. Lett.* **119**, 141101 (2017).
- [41] N. J. Cornish, *Phys. Rev. D* **102**, 124038 (2020).
- [42] J. Ellis and N. Cornish, *Phys. Rev. D* **93**, 084048 (2016).
- [43] B. Bécsy and N. J. Cornish, [arXiv:2011.01942](https://arxiv.org/abs/2011.01942).
- [44] T. Robson and N. J. Cornish, *Phys. Rev. D* **99**, 024019 (2019).
- [45] <https://www.gw-openscience.org>.
- [46] R. Pordes *et al.*, *J. Phys. Conf. Ser.* **78**, 012057 (2007).
- [47] I. Sfiligoi, D. C. Bradley, B. Holzman, P. Mhashikar, S. Padhi, and F. Wurthwein, in *Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering—Volume 02*, CSIE '09 (IEEE Computer Society, Washington, DC, 2009), pp. 428–432.
- [48] HTCondor, <https://htcondor.readthedocs.io/>.
- [49] BayesWave Conda Feedstock, <https://github.com/conda-forge/bayeswave-feedstock/>.
- [50] IGWN Conda Distribution, <https://computing.docs.ligo.org/conda/>.
- [51] CernVM File System, <https://cernvm.cern.ch/portal/filesystem/>.
- [52] R. Pordes, D. Petravick, B. Kramer, D. Olson, M. Livny, A. Roy, P. Avery, K. Blackburn, T. Wenaus, F. Würthwein, I. Foster, R. Gardner, M. Wilde, A. Blatecky, J. McGee, and R. Quick, *J. Phys. Conf. Ser.* **78**, 012057 (2007).
- [53] I. Sfiligoi, D. C. Bradley, B. Holzman, P. Mhashikar, S. Padhi, and F. Wurthwein, in *2009 WRI World Congress on Computer Science and Information Engineering* (IEEE, New York, 2009), Vol. 2, pp. 428–432, <https://ieeexplore.ieee.org/document/5171374>.