Optimization of model independent gravitational wave search for binary black hole mergers using machine learning

T. Mishra⊚, B. O'Brien⊚, V. Gayathri⊚, M. Szczepańczyk⊚, S. Bhaumik⊚, I. Bartos, and S. Klimenko

Department of Physics, University of Florida, PO Box 118440, Gainesville, Florida 32611-8440, USA

The coherent WaveBurst (cWB) search algorithm identifies generic gravitational wave (GW) signals in the LIGO-Virgo strain data. We propose a machine learning (ML) method to optimize the pipeline sensitivity to the special class of GW signals known as binary black hole (BBH) mergers. Here, we test the ML-enhanced cWB search on strain data from the first and second observing runs of Advanced LIGO and successfully recover all BBH events previously reported by cWB, with higher significance. For simulated events found with a false alarm rate less than 1 yr^{-1}, we demonstrate the improvement in the detection efficiency of 26% for stellar-mass BBH mergers and 16% for intermediate mass black hole binary mergers. To demonstrate the robustness of the ML-enhanced search for the detection of generic BBH signals, we show that it has the increased sensitivity to the spin precessing or eccentric BBH events, even when trained on simulated quasicircular BBH events with aligned spins.

I. INTRODUCTION

The detection of the first gravitational wave (GW) signal GW150914 [1] commenced the age of the GW astronomy. Since then, the Advanced LIGO [2] and Advanced Virgo [3] network of detectors have identified 11 GW candidates during the first two observing runs (O1 and O2) [4], 39 GW candidates in the first half of the third observing run (O3a) [5], and provided 20 GW public alerts to EM astronomers during the second half of the third observing run (O3b) [6]. With the improving sensitivity of the GW detector network, it is essential to refine the search algorithms used to detect GW signals.

Coherent WaveBurst (cWB) is an algorithm that searches for excess power in the time-frequency domain to identify GW signals in the LIGO-Virgo strain data [7,8]. Unlike other analysis pipelines which search for binary black hole (BBH) mergers, cWB does not use template waveform models. Instead, the cWB algorithm is model independent, which makes it a valuable tool in the search for poorly modeled or unexpected GW sources. cWB played an integral role in the discovery of the first BBH merger GW150914 [1] and, more recently, in the first direct detection of an intermediate mass black hole (IMBH) GW190521 [9,10]. Also, cWB has contributed to the detection of 22 BBH events in the O1, O2, and O3a observing runs [4,5].

The cWB pipeline generates summary statistics for every identified event. These summary statistics describe generic properties of reconstructed events such as the characteristic frequency, duration, cross-correlation between different detectors, and other parameters described in the Appendix A. In the standard cWB search framework, we use summary statistics to construct vetoes designed to reject noise events from the analysis and increase the significance of detected GW events. Although this veto procedure generally works well, it risks the removal of GW signal events that lie near the border of the predefined veto thresholds. In addition, designing vetoes is challenging since they need to be redefined for each detector network and are dependent on the run conditions.

Machine learning (ML) offers a novel approach to solving complex problems. Accordingly, the interest of ML techniques in GW astronomy has grown in recent years [11–13] as ML has been applied to categorize noise artifacts in the GW detector strain data [14–16], classify GW signals [17–20], and estimate GW source parameters [21–24]. ML has already been used in combination with cWB for various other studies [25–28].

In this paper, we propose to use a decision tree based ensemble learning algorithm called eXtreme-Gradient Boost (XGBoost) [29] to automate the signal-noise classification in cWB and optimize the pipeline sensitivity to BBH mergers. To preserve the waveform independent analysis, we do not attempt to train the ML model directly on the GW strain data. Instead, we utilize cWB to reconstruct events and generate their summary statistics, and then we carefully select a subset of summary statistics used for the construction of the ML model. The end result is the ML-enhanced search pipeline which is resistant to overfitting and provides the robust recovery of GW events with the waveform parameters that could be outside of the training set. We test the cWB pipeline enhanced with

023014-1

XGBoost on publicly available LIGO Hanford and LIGO Livingston strain data from O1 and O2 [30].

The paper is organized as follows. In Sec. II, we introduce the cWB search pipeline. In Sec. III, we describe the data used to train and test our ML algorithm. In Sec. IV, we demonstrate the implementation of ML into the detection procedure and define our new cWB reduced detection statistic. In Sec. V, we compare the sensitivity of the ML-enhanced cWB search against the sensitivity of the standard cWB search. We report the updated significance of BBH events detected during the O1 and O2 observing runs. Finally, in Sec. VI, we state the conclusions of our study.

II. cWB SEARCH ALGORITHM

The cWB search algorithm is designed to detect GW signals with minimal assumptions on the signal model [8,31]. The detector strain data is mapped to the time-frequency domain using the Wilson Daubechies Meyer (WDM) wavelet transform [32] where the data is normalized by the rms of the detector noise. The algorithm then identifies WDM wavelets with excess power above the average fluctuations of the detector noise. The selected nearby wavelets are grouped into clusters. The pipeline generates an event for each selected cluster and reconstructs the signal waveform using the constrained maximum likelihood method [8].

For each event, various summary statistics are estimated by the pipeline which describes the time-frequency structure, signal strength, and coherence across the detector network. The main detection statistic for the cWB generic GW search is the signal-to-noise ratio (SNR) defined for the LIGO detector network as:

$$\eta_0 = \sqrt{\frac{E_c}{2\max(1,\chi^2)}}. \tag{1}$$

Here, E_c denotes the coherent energy estimated by cross-correlating the reconstructed signal waveforms across different detectors, and $\chi^2 = E_n/N_{df}$ where E_n is the estimated residual noise energy and N_{df} is the number of independent wavelet amplitudes describing the event. The χ^2 correction in Eq. (1), which is close to unity for genuine GW events, reduces the non-Gaussian noise contribution. For the cWB searches which target BBH events, the detection statistic is modified to favor events which frequency is increasing with time:

$$\eta_1 = \eta_0 F_M \sqrt{e_M}, \tag{2}$$

where F_M is the event energy fraction and e_M is the event ellipticity defined in Ref. [33]. Both parameters F_M and e_M are close to unity for BBH events and penalize events which time-frequency evolution is significantly different from the chirping BBH signal.

GW detector data is hindered by noise artifacts known as glitches, and consequently, some noise events are reconstructed by the pipeline and leak into the analysis. In the standard cWB analysis, we apply a series of vetoes to target and remove these glitches. This approach, henceforth known as the *veto method*, improves the significance of candidate GW events by reducing excess background. The veto method consists of applying *a priori* defined veto thresholds on a set of the cWB summary statistics. This procedure discretely classifies generated events into one of the two categories: signal-like events and noiselike events. Events that fall into the noiselike category are removed from the analysis. This process could inevitably result in discarding borderline GW events which do not pass the veto thresholds and at the same time makes the pipeline vulnerable to the high SNR glitches which do pass the vetoes. Designing vetoes in the multidimensional space of the summary statistics is challenging, and furthermore, requires retuning of the veto thresholds for each detector network configuration and each observing run.

In the standard cWB setup, the veto method is tuned separately to improve the search sensitivity to stellar-mass BBH mergers ($M_{tot} \lesssim 100\, M_\odot$) and IMBH binary mergers ($M_{tot} \gtrsim 100\, M_\odot$). While the GW waveforms of these two classes are conceptually similar, the corresponding GW signals observed in the LIGO frequency band are quite different. A GW signal originating from the stellar-mass BBH merger usually exhibits the full inspiral-merger-ringdown waveform, whereas GW signals from IMBH binary mergers are short in duration and contain mostly the merger-ringdown waveform, with the inspiral signal buried inside the low-frequency seismic noise. As a result, we utilize two configurations of the cWB search tuned for these systems: the BBH configuration which targets stellar-mass BBH mergers, and the IMBH configuration which targets IMBH binary mergers [10]. IMBH binaries are expected to merge at lower frequencies compared to the stellar-mass BBH mergers, and so the corresponding cWB search configurations apply different vetoes to account for the difference in the signal morphology.

III. DATA

We analyze publicly available strain data from Advanced LIGO's first two observational runs [30]. Here, we only examine data from the LIGO Hanford and LIGO Livingston detectors, with the inclusion of Virgo data left for future work. To train and test our ML model, we require a representative set of noise events and signal events.

Noise events (background) are generated by systematically time-shifting the data from one detector with respect to other detectors in the detector network. Each time shift is chosen to be greater than one second to exclude astrophysical signals. This process is repeated multiple times for various time lags, and we count the number of background events generated over the total accumulated background

time. For the O1 run (GPS time: 1126051217—1137254417), we accumulated approximately 16,000 years and 4,000 years of the background time for the BBH and IMBH searches, respectively. For the O2 run (GPS time: 1164556817—1187740818), we accumulated approximately 11,000 years of background for each search.

To generate a representative set of the signal events, we add (inject) simulated GW signals to the detector data and reconstruct them with cWB. In this work, we investigate four sets of simulated signals: (i) a quasicircular spin-aligned stellar-mass BBH set, (ii) a quasicircular IMBH binary set, (iii) an eccentric BBH set, and (iv) a quasicircular precessing BBH set. Only the first two simulation sets of the quasicircular signals were used for ML training, whereas the remaining two sets are used to test the robustness of the ML implementation. In all four cases, the binary orientation parameters (sky location, inclination angle) for every simulated waveform are randomly drawn from uniform distributions. The redshift z is drawn from a uniform distribution in comoving volume, assuming Planck 2015 cosmology [34].

To simulate the stellar-mass BBH set, we use the SEOBNRv3 [35] and SEOBNRv4 [36] waveform approximants. These waveform approximants include only the dominant ($\ell = 2$, $m = 2$) harmonic mode. The source frame total mass for these simulations ranges from approximately 5 $M_\odot$ to 100 $M_\odot$, and the mass ratio $q = m_2/m_1$ ranges from approximately 1/4 to 1. Component black hole spins are aligned with the orbital plane.

For the IMBH binary set, we use numerical relativity waveforms which include higher-order harmonics. We consider 17 mass bins, as used in Ref. [37], which range in source frame total mass from 120 $M_\odot$ to 800 $M_\odot$, with mass ratios ranging from 1 to 1/10.

For the high mass, eccentric BBH set, we also use numerical relativity waveforms [38,39]. We consider 28 mass bins which range in total mass from 100 $M_\odot$ to 250 $M_\odot$, mass ratio equal to 1, with eccentricities ranging from 0.75 to 0.99.

For the precessing stellar-mass BBH set, we use the SEOBNRv4PHM [40] waveform approximant, which includes precession and higher-order harmonic modes. The source frame total mass ranges from 4 $M_\odot$ to 200 $M_\odot$, with mass ratios ranging from 1 to 1/20. Component black hole spins are isotropically distributed.

The amount of simulation and background data used for training and testing the ML algorithm is described in Table II and further details on the training and testing procedure is explained in Sec. IV B.

IV. MACHINE LEARNING IMPLEMENTATION

The veto method used in the standard cWB search categorizes events into two discrete bins: signal events and noise events. Here, the veto method effectively acts as a decision tree with only two leaves, where the summary statistics for a given event are compared against various rules at each decision node until the event is classified as either a signal event or a noise event. Since this method produces a discrete outcome, it unavoidably removes signal events which do not pass all veto thresholds.

We propose using ML, which produces continuous ranking criteria for all reconstructed events, to replace the veto method. Binary classification is a standard problem in the ML literature. Moreover, many prominent ML algorithms are based on the decision tree structure, which we expect is well suited for the cWB classification problem.

We use the boosted decision tree based ML algorithm called XGBoost [29]. In XGBoost, instead of using a single decision tree to classify events, an ensemble of decision trees is generated. A decision tree is used as the base learner, and subsequent learners (trees) are formed based on the residual errors obtained after each iteration (boosting). This process is expected to be more accurate and more robust than using a single decision tree used by the veto method. A continuous score is calculated by taking the weighted average of output, obtained from each decision tree in the ensemble. The final output P_{XGB} is computed by taking the sigmoid of the continuous score, where a value close to zero denotes a noiselike event and a value close to one denotes a signal-like event.

To construct the ML model, we select a subset of 14 summary statistics estimated by cWB as input features for the ML algorithm. Selected summary statistics describe the signal strength and the correlation across the detectors (η_0, c_c, n_f), the quality of the likelihood fit (E_c/L, χ^2), the time-frequency evolution of the event (ΔT_s, ΔF_s, f_0, $\mathcal{M}$, e_M) and the two different estimators for the number of cycles in the reconstructed waveform (Q_a, Q_p). The detailed list of selected summary statistics and their definitions can be found in Appendix A.

A. Tuning XGBoost hyper-parameters

XGBoost has a number of free hyper-parameters which control various properties of the learning process to prevent overfitting. These hyper-parameters need to be tuned for each specific application. We use a small data set consisting of 200 yr of background data (which equates to approximately 500,000 noise events) and 2,000 simulated IMBH binary events to tune the XGBoost hyper-parameters. These background and simulation events are drawn from the O2 run.

We perform a grid search over a range of six standard XGBoost hyper-parameters, listed in Table I. We find the most optimal set by evaluating each configuration of XGBoost hyper-parameters with respect to the precision-recall area under the curve (AUC PR) metric [41] over 10-fold cross-validation. The optimal configuration of hyper-parameters, according to this criteria, is shown in bold in Table I. We use a method known as *early stopping*

TABLE I.   Entries for XGBoost hyperparameters. We perform a grid search over 432 combinations of hyperparameters. Bold entries indicate the optimal choice.

XGBoost hyper-parameter	Entry
objective	binary:logistic
tree_method	hist
grow_policy	lossguide
n_estimators	20,000[a]
max_depth	7, 9, **11**, 13
learning_rate	0.1, **0.03**
min_child_weight	5.0, **10.0**
colsample_bytree	**0.6**, 0.8, 1.0
subsample	0.4, 0.6, **0.8**
gamma	2.0, **5.0**, 10.0

[a]n_estimators is optimized using early stopping, described in the main text.

to optimize the total number of trees generated. In early stopping, a small fraction of the training data set is set aside for validation. When the validation AUC PR score stops improving, the training ends to prevent the XGBoost from overfitting.

Overall, we found that the performance of the model was not highly dependent on the chosen hyper-parameter values. As a result, we keep this hyper-parameter configuration for all models presented in this paper.

B. XGBoost model training

We train a separate ML model for each search configuration and each observing run. In this paper, we have two search configurations (BBH, IMBH) and analyze two observing runs (O1, O2), a total of four models. The estimated central frequency f_0 of a GW signal is expected to be inversely proportional to the red-shifted total mass of the binary system. As such, we select events with $60\text{ Hz} < f_0 < 300\text{ Hz}$ to train the BBH search models and events with $f_0 < 200\text{ Hz}$ to train the IMBH search models. Each trained model uses the same XGBoost hyper-parameters discussed in Sec. IV A, and consists of around 600 total trees for O1 run and 1000 total trees for O2 run, with an average of 20 leaves per tree.

For training, we select 100 yr of background data (approximately 250,000 noise events) per data chunk. For every 100 yr of background data we select approximately 1000 simulation events. We use simulated quasi-circular stellar-mass BBH mergers (simulation set (i) to train the BBH search model, and we use simulated quasicircular IMBH binary mergers (simulation set (ii) for the IMBH search model. The remaining background and simulation data are used for testing as reported in Table II. The ML algorithm's training and testing procedures are computationally efficient and are completed within minutes using one CPU core. Generally, ML classifiers are expected to be more accurate when the same number of events per

TABLE II.   Amount of background data (given in years) and the number of simulated events used for training and testing the respective ML models. We consider different observing runs (O1, O2), search configurations (BBH, IMBH), and additional simulation cases to test the model robustness (eccentric BBH, precessing BBH).

Dataset	Background [yr]		Simulation	
	Training	Testing	Training	Testing
O1 BBH	900	15316	8662	38246
O1 IMBH	900	3123	8640	40806
O2 BBH	2100	8625	20957	203954
O2 IMBH	2100	8625	20776	150780
Eccentric BBH	…	18000	…	42910
Precessing BBH	…	18000	…	23043

class are used for training. However, in our case, using balanced classes is not feasible. It is difficult to arbitrarily increase the number of simulated events due to the computational cost. Moreover, down-sampling the noise event set is not prudent since we could lose valuable information related to the high SNR tail of the background distribution. Instead, we apply a weight to every sample noise event to reduce the class imbalance. This weight is dependent on η_0 and gives less importance to the low SNR glitches. The weighting procedure is described in more detail in Appendix B.

C. cWB + ML detection statistic

We incorporate the predictions made by the ML model directly into the detection statistic to improve noise rejection. We define the reduced detection statistic used for the ML-enhanced cWB search as:

$$\eta_{\mathrm{r}} = \sqrt{\frac{E_{\mathrm{c}}}{1 + \chi^2(\max(1,\chi^2) - 1)}} W_{\mathrm{XGB}}, \qquad (3)$$

where W_{XGB} is the XGBoost penalty factor. To compute W_{XGB}, we first apply a correction to the XGBoost output, defined in Appendix C 1. This correction is designed to suppress numerous noise events which have less than one cycle in the time domain waveform, which is typical for the known family of glitches found in the GW detector data. Next, we apply a monotonic transformation, defined in Appendix C 2, to obtain the penalty factor $W_{\mathrm{XGB}} = W_{\mathrm{XGB}}(P_{\mathrm{XGB}})$. This transformation accentuates the ranking of events with the values of P_{XGB} very close to unity. Although W_{XGB} itself could be used as a detection statistic, we find that it is susceptible to assigning high significance to low SNR noise events. Instead, we use it as a penalty factor to the estimated effective correlated SNR η_0. The end result is a detection statistic η_{r} which is enhanced by the ML classification and resistant to overfitting the low SNR noise events.

V. RESULTS

In this section, we present the results of the ML-enhanced cWB search and compare its sensitivity to the standard cWB search. The significance of a given candidate event is estimated by its false alarm rate (FAR). It is computed by counting the number of background events with the equal or higher value of the detection statistic than for the candidate event, divided by the total background time.

For the ML-enhanced cWB search, FAR is calculated by using the reduced detection statistic η_r [Eq. (3)]. Whereas for the standard cWB BBH search, we first remove vetoed events and then calculate FAR by using the η_1 detection statistic [Eq. (2)]. We compare the performance of the different cWB searches by comparing their detection efficiency defined as the number of detected simulated events with FAR equal to or less than a given threshold, divided by the total number of recovered simulation events.

A. Reanalysis of O1 and O2 data

First, we examine the background noise distributions for the ML-enhanced search. Figure 1 shows the rate of the noise events for each search (BBH, IMBH) and each observing run (O1, O2) as a function of the reduced detection statistic η_r. In all cases, we do not observe significant tails of the high η_r events, which indicates that the ML-enhanced detection statistic efficiently suppresses the high SNR outliers.

Next, we examine the sensitivities of each search for various simulation data sets. The top panel of Fig. 2 shows the detection efficiency as a function of FAR for the various searches over the O1 (left) and O2 (right) data. The ML-enhanced search (shown in red) is more sensitive than the standard search with vetoes (black) in the wide range of the FAR thresholds. For events detected with FAR < 1 yr^{-1},
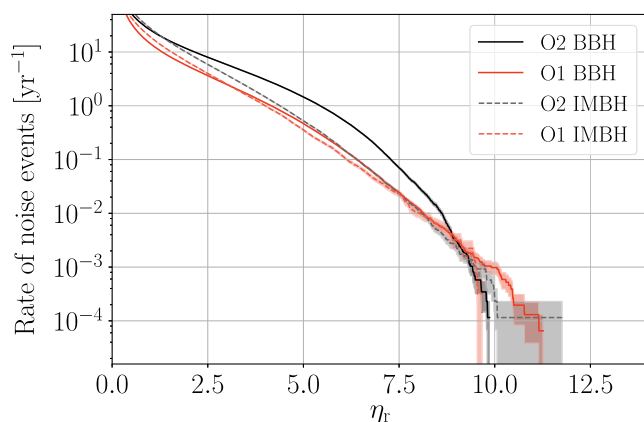

FIG. 1. Rate of noise events vs detection statistic η_r for each search configuration (BBH, IMBH) and each observing run (O1, O2). We do not observe significant tails in the η_r distributions.

we estimate the 26% improvement for the BBH configuration and the 16% improvement for the IMBH configuration, averaged over the two observing runs. For high significance detection (FAR < 100 yr^{-1}), we estimate the 22% improvement for the BBH configuration and the 13% improvement for the IMBH configuration.

The bottom panel of Fig. 2 shows the detection efficiency for events detected with FAR < 1 yr^{-1} as a function of the central frequency f_0. Here, we see that for most frequency bins, the ML-enhanced search (red) is more sensitive than the standard search (black). This indicates that the ML-enhanced search is not overly tuned to any specific frequency bins but shows consistent improvement over the entire frequency range considered by the cWB search configurations. Since the central frequency f_0, is expected to be inversely proportional to the detector frame mass, we can infer that the ML-enhanced search is more sensitive over the entire BBH mass range accessible by LIGO.

Table III reports the BBH candidates identified by the ML-enhanced cWB search in the O1-O2 observing runs. We recover 7 BBH candidates previously reported by the standard cWB search [4], all identified with higher significance. Additionally, the ML-enhanced cWB search detected GW170809 with a FAR of 0.29 yr^{-1}, which was previously vetoed in the standard cWB search. No other candidate events were identified with FAR less than 1 yr^{-1}.

B. Test of model robustness

As a final test, we analyze the performance of the ML-enhanced search on simulated waveforms outside of the training set. First, we investigate the sensitivity of the ML-enhanced IMBH search, which is trained on quasi-circular binaries, to high mass BBH systems with highly eccentric orbits (simulation set iii). In Fig. 3, we show the sensitivity to the eccentric IMBH mergers of the ML-enhanced IMBH search compared to the standard IMBH search. The ML-enhanced search is more sensitive to the eccentric BBH mergers despite the fact that we trained the ML model only on the quasi-circular IMBH waveforms.

Next, we test the ML-enhanced BBH search on precessing BBH systems (simulation set iv). The training set consists of simulated waveforms with the (2,2) harmonic mode, aligned spins, and low mass ratio (1–1/4). The testing set consists of simulated waveforms with higher-order modes, precessing spins, and a higher mass ratio (1–1/20). In this case, the model is also trained on O2 data, whereas the testing set consists of O3a simulation and background, which has very different GW detector sensitivity. Figure 4 shows improved detection efficiency of the ML-enhanced BBH search compared to the standard BBH search.
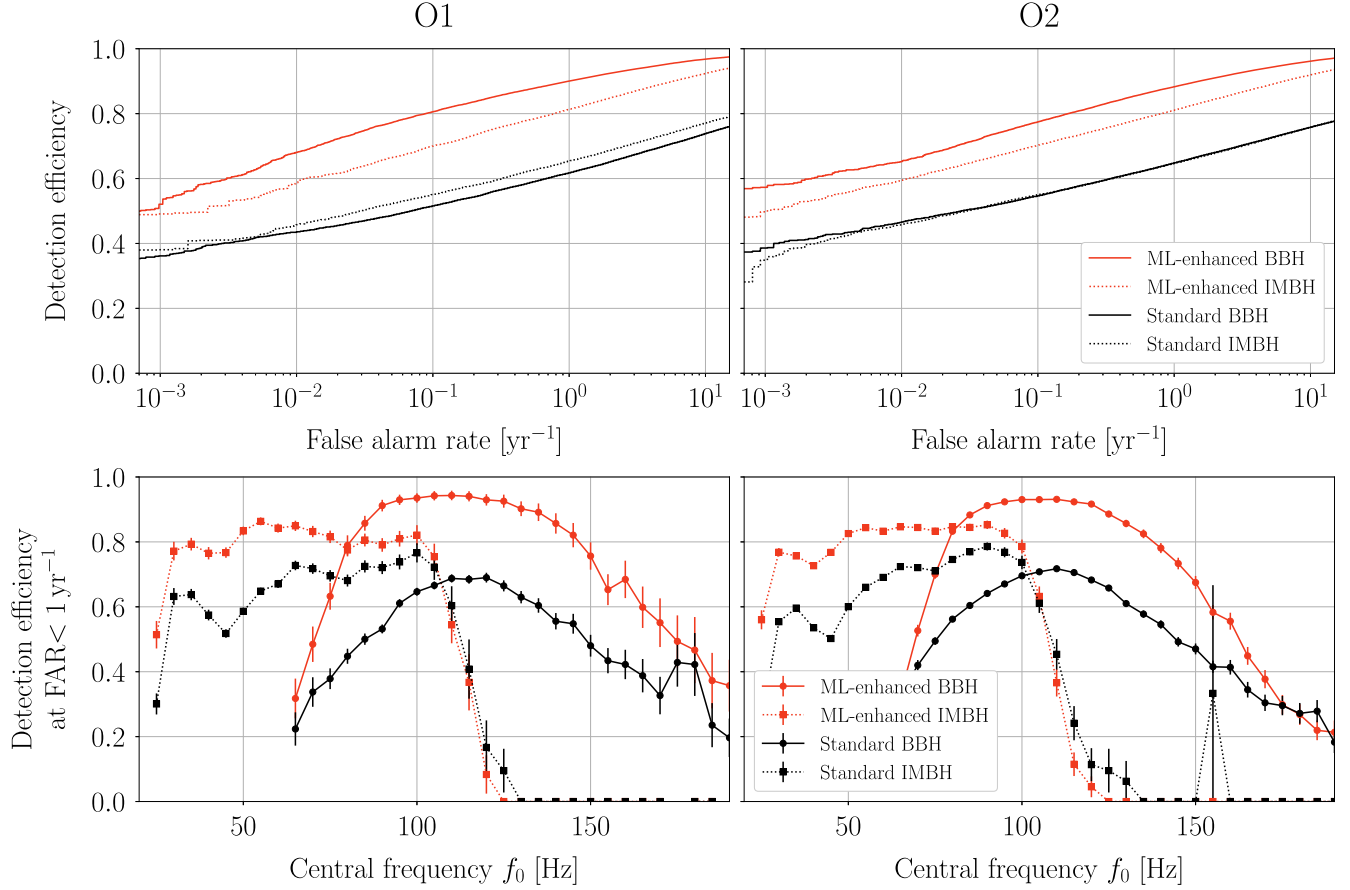
FIG. 2. Top: detection efficiency vs FAR for the O1 observing run (left) and the O2 observing run (right). Bottom: detection efficiency for events identified with FAR less than 1 yr^{-1} as a function of the central frequency f_0 for O1 (left) and O2 (right). Solid lines correspond to the BBH configuration, while dotted lines represent the IMBH configuration. Red curves represent the sensitivity of the ML-enhanced cWB search, and black curves represent the sensitivity of the standard cWB search with the veto method.

These results demonstrate that the ML-enhanced search is agnostic to the details of the BBH dynamical evolution including eccentricity, spin effects, and higher-order modes. It retains the robustness of the standard search, but with increased sensitivity.

TABLE III. O1-O2 event candidates detected with the standard cWB BBH search (η_1 + vetoes), and the ML-enhanced cWB BBH search (η_r). We report all detections with FAR less than 1 yr^{-1}. Entries with a "<" sign indicate that the estimated significance is limited by the amount of available background data.

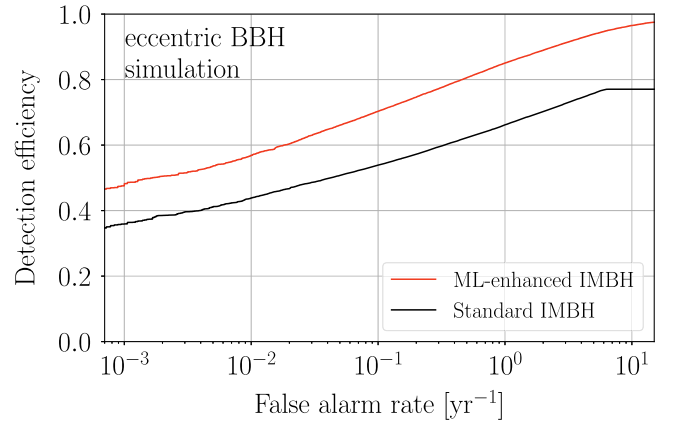| Event	Standard cWB (η_1 + vetoes) FAR [yr^{-1}]	ML-enhanced cWB (η_r) FAR [yr^{-1}]
GW150914	$<1.6 \times 10^{-4}$	$<7 \times 10^{-5}$
GW151226	2×10^{-2}	6.5×10^{-3}
GW170104	2.9×10^{-4}	$<1.2 \times 10^{-4}$
GW170608	1.4×10^{-4}	$<1.2 \times 10^{-4}$
GW170729	2×10^{-2}	$<1.2 \times 10^{-4}$
GW170809	$\cdots$	2.9×10^{-1}
GW170814	$<2.1 \times 10^{-4}$	$<1.2 \times 10^{-4}$
GW170823	2.1×10^{-3}	$<1.2 \times 10^{-4}$

FIG. 3. Detection efficiency vs FAR for a high mass, eccentric BBH simulation set recovered with the cWB IMBH configuration. The red curve represents the sensitivity of the ML-enhanced cWB search, and the black curve represents the sensitivity of the standard cWB search.
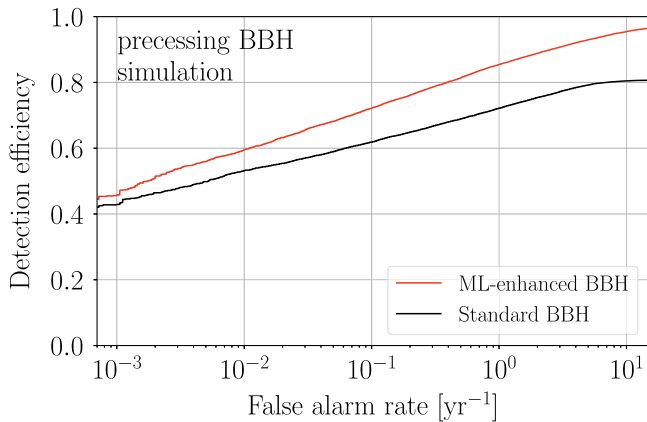
FIG. 4. Detection efficiency vs FAR for a precessing BBH simulation set recovered with the cWB BBH configuration. The red curve represents the sensitivity of the ML-enhanced cWB search, and the black curve represents the sensitivity of the standard cWB search.

VI. CONCLUSION

In conclusion, we introduce a novel method to automate the noise rejection in cWB with ML, and we demonstrate that the ML-enhanced cWB search has improved sensitivity to the BBH mergers. Unlike the standard cWB search, which decimates noise events with the veto method, the ML-enhanced search uses a continuous ranking statistic and does not remove any events. Instead, it penalizes noise events with the newly designed detection statistic η_r, which combines the correlated network SNR with the penalty factor based on the signal-noise classification provided by the XGBoost model. This new detection statistic η_r is resistant to loud noise glitches and allows cWB identification of BBH events with higher significance.

For the stellar-mass BBH mergers, the detection efficiency of the ML-enhanced cWB search is improved by 26% compared to the standard cWB search (at FAR less than 1 yr^{-1}). Similarly, for the simulation set of IMBH binary mergers, the detection efficiency is improved by 16%. While we do not claim any new BBH detections in the O1 and O2 data, we do improve the detection confidence of the previously identified GW candidates and recovered the GW170809 event missed by the standard cWB search.

The ML-enhanced cWB search is capable of detecting BBH signals well outside of the training set. We demonstrate the improved pipeline sensitivity to the highly eccentric BBH mergers despite only being trained on the quasicircular IMBH signals. We also found the search to be agnostic to other binary waveform properties including precession, high mass ratio, and higher-order harmonic modes. We expect the search to be sensitive to BBH signals even if the GW waveforms used for training are not exactly representative of true astrophysical signals.

The ML-enhanced BBH search is a promising addition to the cWB pipeline for future planned observing runs, where we expect numerous BBH detections. While in this study, we use only the LIGO Hanford and LIGO Livingston detector network, in future work we will expand the ML-enhanced search to the other detector networks which include Virgo and Kagra detectors. This will further improve the cWB sensitivity to the BBH mergers. The ML-enhanced pipeline could be also used for the low latency cWB searches in the future observing runs.

APPENDIX A: cWB ESTIMATED SUMMARY STATISTICS

The ML algorithm is tuned and trained on the data from a selected subset of cWB summary statistics for each event. We start by truncating this subset by selecting the summary statistics that have a low correlation with each other. We also aggregate a few summary statistics together to prune the list of summary statistics that are used as input features for the ML algorithm. We select 14 cWB summary statistics in total that are used by the ML algorithm as the input list of features. The summary statistics are listed below:

(i) η_0—Main cWB detection statistic for the generic GW search. For the ML study, we cap the η_0 value at 8 (any event with higher η_0 is assigned a value of 8) so the algorithm is not affected by the high SNR events in the background distribution, which is a steep function of η_0.

(ii) c_c—Coherent energy divided by the sum of coherent energy and null energy, defined in Ref. [31].

(iii) n_f—Effective number of time-frequency resolutions used for event detection and waveform reconstruction.

(iv) E_c/L—Ratio of the coherent energy to the network likelihood.

(v) ΔT_s—Energy weighted signal duration.

(vi) ΔF_s—Energy weighted signal bandwidth.

(vii) f_0—Energy weighted signal central frequency.

(viii) $\mathcal{M}$—Chirp mass parameter estimated in the time-frequency domain, defined in Ref. [33].

(ix) e_M—Chirp mass goodness of the fit metric, presented in Ref. [33].

(x) Q_a—The waveform shape parameter [46] developed to identify a characteristic family of (blip) glitches present in the detectors [4,47]. A value of Q_a corresponds to the cWB event being a blip glitch.

(xi) Q_p—An estimation of the effective number of cycles in a cWB event.

(xii) L_v—for the loudest pixel, the ratio between the pixel energy and the total energy of an event [48].

(xiii) χ^2—quality of the event reconstruction, $\chi^2 = E_n/N_{df}$ where E_n is the residual noise energy estimated and N_{df} is the number of independent wavelet amplitudes describing an event.

(xiv) C_n—Data chunk number. LIGO-Virgo data is divided into time segments known as chunks, which typically contain a few days of strain data. Including the data chunk number allows the ML algorithm to respond to changes in detector sensitivity across separate observing runs and chunks.

APPENDIX B: NOISE EVENT SAMPLE WEIGHT

In the initial testing phase with the trained XGBoost model, we found a tail of high SNR background events, most of which were consistent with blip glitches. This tail is caused by the suboptimal ML model due to the high class imbalance between the high SNR noise events and signal events. To correct for this tail, we applied an η_0 dependent sample weight to the noise events while training the XGBoost models. This weight provides us with a weighted background distribution as shown in Fig. 5, which is similar for any observing run.

The sample weight for the simulation events is set to 1. For the noise events, we divide η_0 into 231 bins with values

ranging from 5 to 8 and apply the sample weight w_B for each bin as follows:

$$w_B(\eta_0) = \frac{(N_S(\eta_0) - a) + be^{c(d-\eta_0)}}{N_B(\eta_0)}, \qquad (B1)$$

where $N_S(\eta_0)$ is the number of signal events and $N_B(\eta_0)$ is the number of noise events, in a given bin. We found that the sample weight works reasonably well for the following choices of values: $a = 65$, $b = 0.8$, $c = 0.9$, and $d = 12.9$. The number of simulation events for $\eta_0 \geq 8$ were resampled to match the number of background events with the same range of η_0. This application of sample weight enables the algorithm to successfully remove almost all the high SNR outliers while keeping the weighted events class imbalance (N_B/N_S) to around 20 for any given observing run. Without the application of the sample weight, the class imbalance ranges from 50 to 600 or even higher for the given training setup depending on the observing run and the search configuration.

APPENDIX C: XGBOOST PENALTY FACTOR

1. cWB correction to P_{XGB} in the IMBH configuration

Further investigation revealed that in the IMBH configuration, high SNR background outliers were found in a specific parameter space represented by the $Q_a - Q_p$ plot in Fig. 6. In the standard veto method, the application of the Q_a and Q_p vetoes would have removed all events below the predefined thresholds at the cost of losing a small fraction of simulated events. In the ML method, we apply a correction to the ranking criteria P_{XGB} such that we suppress the background outliers in the affected parameter space below the magenta-colored curve in Fig. 6.

FIG. 6. $Q_a - Q_p$ parameter space plot for the IMBH configuration with O1 run data. Black dots represent loud background noise events ($\eta_0 \geq 8$). Red dots represent loudest background outliers ($\eta_0 \geq 10$). The heat map represents the simulated IMBH distribution.
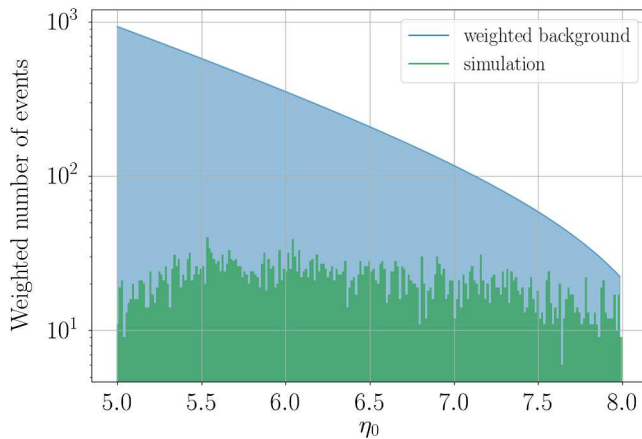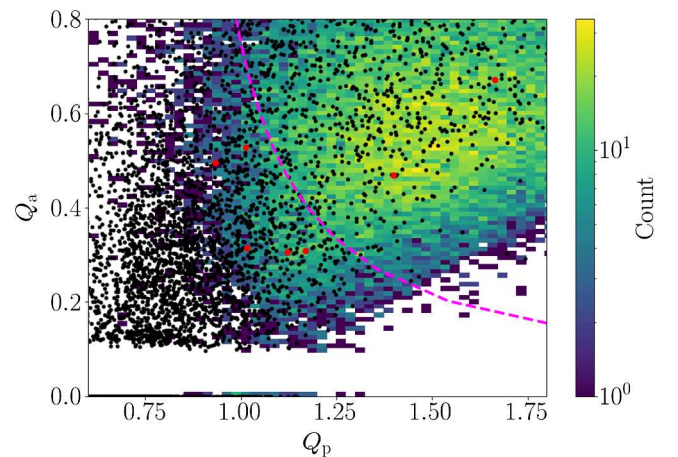

FIG. 5.   Weighted event distribution as a function of η_0 for background events and simulated events. Here, we show the IMBH configuration with the O1 run data.

The correction to the continuous ranking criteria output P_{XGB} from XGBoost is applied only to the IMBH configuration (shorter signals with fewer cycles in the time domain), given by the following equations:

$$P_{\mathrm{XGB}} = \begin{cases} P_{\mathrm{XGB}} - (0.15 - Q_{\mathrm{a}}(Q_{\mathrm{p}} - 0.8)), \\ \quad \text{if } Q_{\mathrm{a}}(Q_{\mathrm{p}} - 0.8) \leq 0.15 \\ \quad \text{(under the curve)} \\ P_{\mathrm{XGB}}, \\ \quad \text{if } Q_{\mathrm{a}}(Q_{\mathrm{p}} - 0.8) > 0.15 \\ \quad \text{(above the curve)}. \end{cases} \quad \text{(C1)}$$

This correction enhances detection of signal and suppresses glitches in the desired part of the $Q_{\mathrm{a}} - Q_{\mathrm{p}}$ parameter space explicitly without any changes to the XGBoost training and testing procedure. The correction comes at the cost of losing a small fraction of IMBH signals with small values of Q_{p}.

2. Monotonic transformation W_{XGB}

The monotonic transformation applied on P_{XGB} is defined as follows,

$$W_{\mathrm{XGB}}(P_{\mathrm{XGB}}) = \frac{-\log(1.0 - 0.995\sqrt{P_{\mathrm{XGB}}})}{5.3}, \quad \text{(C2)}$$

which counterbalances the steep sigmoid function used by XGBoost while producing P_{XGB}. The P_{XGB} output by XGBoost for any event has a precision of up to 5 decimal places. As a result, the effect of using P_{XGB} directly as a penalty factor is not effective. The transformation in Eq. (C2) allows us to zoom in on the high precision P_{XGB} values close to unity. This helps us differentiate between high SNR background events and simulation events that typically end up with high values of P_{XGB}.

[1] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Phys. Rev. Lett. **116**, 061102 (2016).
[2] J. Aasi *et al.* (LIGO Scientific Collaboration), Classical Quantum Gravity **32**, 074001 (2015).
[3] F. Acernese *et al.* (Virgo Collaboration), Classical Quantum Gravity **32**, 024001 (2015).
[4] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Phys. Rev. X **9**, 031040 (2019).
[5] R. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Phys. Rev. X **11**, 021053 (2021).
[6] Gravitational-Wave Candidate Event Database, LIGO/Virgo Public Alerts (2020).
[7] S. Klimenko, I. Yakushin, A. Mercer, and G. Mitselmakher, Classical Quantum Gravity **25**, 114029 (2008).
[8] S. Klimenko *et al.*, Phys. Rev. D **93**, 042004 (2016).
[9] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Phys. Rev. Lett. **125**, 101102 (2020).
[10] M. Szczepanczyk, S. Klimenko, B. O'Brien, I. Bartos, V. Gayathri, G. Mitselmakher, G. Prodi, G. Vedovato, C. Lazzaro, E. Milotti, F. Salemi, M. Drago, and S. Tiwari, Phys. Rev. D **103**, 082002 (2021).
[11] E. Cuoco *et al.*, Mach. Learn. Sci. Technol. **2**, 011002 (2020).
[12] E. A. Huerta *et al.*, Nat. Rev. Phys. **1**, 600 (2019).
[13] E. A. Huerta and Z. Zhao, arXiv:2105.06479.
[14] R. E. Colgan, K. R. Corley, Y. Lau, I. Bartos, J. N. Wright, Z. Márka, and S. Márka, Phys. Rev. D **101**, 102003 (2020).
[15] M. Zevin, S. Coughlin, S. Bahaadini, E. Besler, N. Rohani, S. Allen, M. Cabero, K. Crowston, A. K. Katsaggelos, S. L. Larson *et al.*, Classical Quantum Gravity **34**, 064003 (2017).

[16] E. Cuoco, M. Razzano, and A. Utina, in *2018 26th European Signal Processing Conference (EUSIPCO)* (2018), pp. 2648–2652, https://ieeexplore.ieee.org/document/8553393.
[17] T. D. Gebhard, N. Kilbertus, I. Harry, and B. Schölkopf, Phys. Rev. D **100**, 063015 (2019).
[18] A. Iess, E. Cuoco, F. Morawski, and J. Powell, Mach. Learn. Sci. Technol. **1**, 025014 (2020).
[19] D. George and E. A. Huerta, Phys. Rev. D **97**, 044039 (2018).
[20] H. Gabbard, M. Williams, F. Hayes, and C. Messenger, Phys. Rev. Lett. **120**, 141103 (2018).
[21] D. George and E. Huerta, Phys. Lett. B **778**, 64 (2018).
[22] S. Schmidt, M. Breschi, R. Gamba, G. Pagano, P. Rettegno, G. Riemenschneider, S. Bernuzzi, A. Nagar, and W. Del Pozzo, Phys. Rev. D **103**, 043020 (2021).
[23] S. R. Green and J. Gair, arXiv:2008.03312.
[24] H. Shen, E. A. Huerta, E. O'Shea, P. Kumar, and Z. Zhao, arXiv:1903.01998.
[25] S. Vinciguerra, M. Drago, G. A. Prodi, S. Klimenko, C. Lazzaro, V. Necula, F. Salemi, V. Tiwari, M. C. Tringali, and G. Vedovato, Classical Quantum Gravity **34**, 094003 (2017).
[26] M. Cavaglià, S. Gaudio, T. Hansen, K. Staats, M. Szczepańczyk, and M. Zanolin, Mach. Learn. Sci. Technol. **1**, 015005 (2020).
[27] V. Gayathri, D. Lopez, P. R. S., I. S. Heng, A. Pai, and C. Messenger, Phys. Rev. D **102**, 104023 (2020).
[28] B. O'Brien, M. Szczepańczyk, V. Gayathri, I. Bartos, G. Vedovato, G. Prodi, G. Mitselmakher, and S. Klimenko, arXiv:2106.00605.
[29] T. Chen and C. Guestrin, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge*

Discovery and Data Mining, KDD '16 (Association for Computing Machinery, New York, NY, USA, 2016), p. 785–794, https://doi.org/10.1145/2939672.2939785.

[30] R. Abbott *et al.*, SoftwareX **13**, 100658 (2021).

[31] S. Klimenko, I. Yakushin, A. Mercer, and G. Mitselmakher, Classical Quantum Gravity **25**, 114029 (2008).

[32] V. Necula, S. Klimenko, and G. Mitselmakher, J. Phys. Conf. Ser. **363**, 012032 (2012).

[33] V. Tiwari, S. Klimenko, V. Necula, and G. Mitselmakher, Classical Quantum Gravity **33**, 01LT01 (2016).

[34] P. A. R. Ade *et al.*, Astron. Astrophys. **594**, A13 (2016).

[35] S. Babak, A. Taracchini, and A. Buonanno, Phys. Rev. D **95**, 024010 (2017).

[36] A. Bohé *et al.*, Phys. Rev. D **95**, 044028 (2017).

[37] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Phys. Rev. D **100**, 064064 (2019).

[38] J. Healy, C. O. Lousto, Y. Zlochower, and M. Campanelli, Classical Quantum Gravity **34**, 224001 (2017).

[39] V. Gayathri, J. Healy, J. Lange, B. O'Brien, M. Szczepanczyk, I. Bartos, M. Campanelli, S. Klimenko, C. Lousto, and R. O'Shaughnessy, arXiv:2009.05461.

[40] S. Ossokine *et al.*, Phys. Rev. D **102**, 044055 (2020).

[41] J. Davis and M. Goadrich, in *Proceedings of the 23rd International Conference on Machine Learning—ICML 06, The campus of Carnegie Mellon University in Pittsburgh, Pennsylvania* (ACM Press, 2006), https://doi.org/10.1145/1143844.1143874.

[42] C. R. Harris *et al.*, Nature (London) **585**, 357 (2020).

[43] J. Reback *et al.*, pandas-dev/pandas: Pandas 1.2.5 (2020), https://zenodo.org/record/5013202#.YN67iOhKiUk.

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

[45] F. Pedregosa *et al.*, J. Mach. Learn. Res. **12**, 2825 (2011).

[46] G. Vedovato, The Qveto algorithm (2018), https://gwburst.gitlab.io/documentation/latest/html/faq.html?highlight=qveto#the-qveto.

[47] J. McIver, Classical Quantum Gravity **29**, 124010 (2012).

[48] G. Vedovato, The Lveto algorithm (2018), https://gwburst.gitlab.io/documentation/latest/html/faq.html?highlight=lveto#the-lveto.