Incorporating information from LIGO data quality streams into the PyCBC search for gravitational waves

Derek Davis, Max Trevor, Simone Mozzon, and Laura K. Nuttall

Phys. Rev. D **106**, 102006 — Published 22 November 2022

DOI: 10.1103/PhysRevD.106.102006

Incorporating information from LIGO data quality streams into the PyCBC search for gravitational waves

Derek Davis[1], Max Trevor[2], Simone Mozzon[3], Laura K. Nuttall[3]
[1] *LIGO, California Institute of Technology, Pasadena, CA 91125, USA*
[2] *University of Maryland, College Park, MD 20742, USA and*
[3] *University of Portsmouth, Portsmouth, PO1 3FX, UK*
(Dated: October 26, 2022)

We present a new method which accounts for changes in the properties of gravitational-wave detector noise over time in the PyCBC search for gravitational waves from compact binary coalescences. We use information from LIGO data quality streams that monitor the status of each detector and its environment to model changes in the rate of noise in each detector. These data quality streams allow candidates identified in the data during periods of detector malfunctions to be more efficiently rejected as noise. This method allows data from machine learning predictions of the detector state to be included as part of the PyCBC search, increasing the total number of detectable gravitational-wave signals by up to 5%. When both machine learning classifications and manually-generated flags are used to search data from LIGO-Virgo's third observing run, the total number of detectable gravitational-wave signals is increased by up to 20% compared to not using any data quality streams. We also show how this method is flexible enough to include information from large numbers of additional arbitrary data streams that may be able to further increase the sensitivity of the search.

I. INTRODUCTION

In the years since the first detection of gravitational waves by LIGO-Virgo [1–3], the rate of detection has grown by over an order of magnitude [4–7]. However, identifying gravitational waves in the collected data still requires the use of analysis pipelines that carefully look through the data. To date, all events detected by LIGO have been identified by at least one pipeline that uses matched filtering [8–10]. A wide variety of matched filter pipelines have been developed to analyze LIGO data [11–14]. One such pipeline that has been in use since the first detection of gravitational waves utilizes the `PyCBC` software suite [15]. We refer to the offline search configuration of this pipeline [16–18] as `PyCBC`.

The `PyCBC` search for gravitational waves from compact binary coalescenses (CBCs), is one of the main matched filter searches used to identify signals in LIGO data. `PyCBC` has been used to identified the vast majority of gravitational-wave signals to date [4–7, 19–21]. As a matched filter search pipeline, `PyCBC` uses templates based on post-Newtonian and numerical models of gravitational-wave signals [22, 23] to identify similar features in gravitational-wave detector strain data. Peaks in the matched filter signal-to-noise ratio (SNR) time series (referred to as "triggers") are found from these templates, and coincident sets of these triggers are assigned a ranking statistic that captures how likely it is that each trigger is a candidate gravitational-wave signal. The significance of these candidates is then estimated by simulated large amounts of background data by shifting the time stamp of triggers in one detector more than the gravitational-wave travel time between each site [24].

One of the main challenges to detecting gravitational waves with matched filter searches is the presence of non-Gaussian noise artifacts in the data. These artifacts are bursts of excess power that are referred to as "glitches." Glitches are problematic for searches for gravitational waves as they can mimic or mask some features of astrophysical signals. It is also known that specific glitches can impact the measured search background [25, 26]. As glitches are known to not be astrophysical in origin, it is imperative that `PyCBC` does not mistake a glitch for a real signal. Numerous features in `PyCBC` are designed to better differentiate glitches from gravitational-wave signals using the gravitational-wave strain data alone. However, any additional information that can better help `PyCBC` differentiate signals from glitches will improve the ability of `PyCBC` to identify gravitational-wave events.

In recent observing runs, hundreds of thousands of additional data streams beyond the gravitational-wave strain data were recorded at each LIGO observatory; these data streams were used to both operate the detector and monitor the detector environment [27, 28]. These additional data streams, referred to as "auxiliary data," can also be used to identify the source of glitches in LIGO data or predict the presence of a glitch in the strain data. Auxiliary data has been shown to be beneficial for use in gravitational-wave analyses to reject potential candidates due to noise [29], subtract contributions of persistent noise from the strain time series [30, 31], and validate the astrophysical origin of observed gravitational-wave events [32]. Information from auxiliary data streams is generally re-packaged into more informative "data quality products" that are used by `PyCBC`.

In this work, we introduce a new method to incorporate information from these data quality products into the `PyCBC` search for gravitational waves. This method is designed to use information from data quality products while evaluating the significance of a given gravitational wave candidate, rather than simply rejecting candidates that occur during data quality flag segments. This is ac-

complished by using data quality information as a part of the statistic used to rank candidates in PyCBC. We will also demonstrate how this method is generic enough to allow any data stream to be incorporated into the search pipeline, including the iDQ time series [33], data quality flags [29, 32, 34], and the large number of auxiliary data streams that are recorded at each site. While the methods described in this text are generalizable to the analysis of data from all ground-based gravitational interferometers, such as Virgo [2] or KAGRA [35], we exclusively work with data from the two LIGO detectors, LIGO Hanford and LIGO Livingston, in this work. We find that use of this new method increases the number of detectable gravitational-waves in a variety of different applications.

This work is organized as follows. In the remainder of this section, we outline the current methods used in the PyCBC search for compact binaries [16–18], with emphasis on how the properties of the detector noise are modelled in the search. We also discuss some of the current products that are produced by the LIGO collaboration to track the data quality. We then explain, in Section II, our proposed improvement to the noise model in PyCBC, and how this improved model can be used in a variety of cases. We demonstrate the benefits of our improved model for multiple applications in Section III. Finally, we discuss how this improved model will benefit future searches for gravitational waves in Section IV.

A. Identifying Gravitational-wave Signals

The PyCBC search for compact binaries [16–18] identifies gravitational-wave events using matched filtering with gravitational waveforms predicted by general relativity [22, 23]. The SNR for a matched filter with a specific waveform template h is [10]

$$\rho^2(t) \equiv \frac{\| \langle s|h\rangle \|^2}{\langle h|h\rangle} \ , \tag{1}$$

where the inner product, $\langle|\rangle$, is defined as

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

with s the strain data, h the template, and $S_n(f)$ the estimated power spectral density for the time in question. This is equivalent to cross-correlation in the frequency domain. Peaks in this SNR time series are labelled as triggers and correspond to potential signals in the data that are similar to the template.

If data from gravitational-wave detectors were purely stationary and Gaussian noise, the matched filter SNR would be sufficient to identify signals in the data. However, variations in the properties of the noise, both over short and long periods, complicate the problem. To account for the non-idealized features in the data, a "ranking statistic" is created that includes additional information about the data and expected signal properties to better differentiate signals from noise. Ideally, an astrophysical signal should receive a high ranking statistic, while a noise fluctuation should receive a lower ranking statistic (often referred to as being 'down-ranked").

The generic form of the ranking statistic is given by the ratio of the signal and noise distributions [18] for a given set of parameters, $\vec{\kappa}$,

$$\Lambda_{opt}(\vec{\kappa}) = \eta_S \frac{\hat{r_S}(\vec{\kappa})}{r_N(\vec{\kappa})} \ , \tag{3}$$

where η_S is the overall rate of signals and $\hat{r_S}(\vec{\kappa})$ is the transfer function between the true rate of signals the detectable rate, and $r_N(\vec{\kappa})$ is the rate of noise.

It is convenient to consider the ratio of these two distributions as the difference of their logarithms,

$$R(\vec{\kappa}) = \log r_S(\vec{\kappa}) - \log r_N(\vec{\kappa}) \ . \tag{4}$$

The parameters, $\vec{\kappa}$, that are used in both the PyCBC signal and noise models encode details about the physical properties of the triggers and how well the measured data matches that expected of an astrophysical signal. In addition to the matched filter SNR, numerous signal consistency tests are included [10, 36] to measure how well a candidate trigger matches the expected signal morphology. The signal model is based on the expected distribution of these parameters for astrophysical signals. These parameters are then used to calculate a single reweighted SNR, $\hat{\rho}$, that quantifies how well the data matches a real signal in each detector. This value is generally referred to as the "single-detector statistic." Additional parameters that corresponds to relationships between the data in multiple detectors are also used. Full details of the signal model are provided in [18]. We will outline the noise model in additional depth for convenience.

The PyCBC noise model is based on fitting the distribution of triggers in the data to an exponential decay function. The rate of noise triggers for a given template in a particular detector, $\vec{\theta}$, is fit to an exponential given by

$$r_N(\hat{\rho};\vec{\theta}) = \mu(\vec{\theta})\alpha(\vec{\theta}) \exp\left[-\alpha(\vec{\theta})(\hat{\rho} - \hat{\rho}_{th})\right] \ . \tag{5}$$

For a given template, $\vec{\theta}$, the term $\mu(\vec{\theta})$ is the number of triggers above threshold and $\alpha(\vec{\theta})$ is the exponential decay rate with respect to $\hat{\rho}$. This $\hat{\rho}$ is the same single-detector ranking statistic that is used in the signal model. Only triggers with $\hat{\rho} > \hat{\rho}_{th}$ are considered in this fit.

In this time-independent PyCBC noise model, a number of approximations are already used to calculate $\mu(\vec{\theta})$ and $\alpha(\vec{\theta})$ [17]. First, a maximum likelihood fit of this

noise model is performed for for each detector and each template in the search individually. However, there are not enough triggers identified per template to to accurately measure both parameters for each template. Kernel smoothing is used to reduce noise in the measured values of μ and α with respect to the duration of each template, $\tau(\vec{\theta})$. Hence the noise model used in the search is $r_N(\hat{\rho};\vec{\theta}) \approx r_N(\hat{\rho};\tau(\vec{\theta}))$.

This model of the noise does not include any time dependence, meaning that this fit assumes a single distribution is valid for each template during the entire analysis period. To account for variation in the properties of the noise with respect to time, it is typical for the PyCBC search to be run separately over relatively short chunks of data (typically 5 days). However, it is known [32, 34] that gravitational-wave detector data contains short term fluctuations on both the hour- and second-scale. While some techniques have been developed to account for these fluctuations [37, 38], they do not introduce explicit time-dependence into the noise model itself.

B. LIGO Data Quality Information

At each LIGO observatory, hundreds of thousands of data streams are recorded during an observing run to control and monitor the detectors [28]. A subset of these data streams have been found to be highly correlated with periods of excess noise in LIGO strain data. For example, ground motion that introduces additional motion of the test masses and increases the chances of scattered light is well monitored by seismic sensors. However, when this information is used to support the astrophysical analyses, this data is first curated into data quality products. These data quality products combine multiple data streams into a single data product that is simpler for astrophysical analyses to utilize. Similar data quality products are produced for other gravitational-wave observatories [39, 40]. The process of developing and finalizing these curated data quality products has taken multiple months in previous observing runs [41], increasing the total amount of time required to complete end-to-end analyses of LIGO data.

One example of a data quality product are "data quality flags" [29, 32, 34], which mark time periods likely to contain glitches based on information from specific auxiliary data streams. Data quality flags are binary data streams sampled at 1 Hz that have multiple categories to indicate the different severity of noise likely to be present in the detector. These flags are used by PyCBC to remove times from an analysis or veto triggers identified during a data quality flag. Other searches [12, 42] instead use these flags to replace the data with zeroes during flagged times.

Another data quality product that has been used in analyses is the iDQ time series [33]. This product is based on a machine-learning algorithm that uses a large number of auxiliary data streams to predict the likelihood of a glitch being present in the detector strain data at a given time. In O3, iDQ was a single time series sampled at 128 Hz. One key difference compared to data quality flags is that iDQ is not a binary data stream, and instead assigns a likelihood to each sample based on the probability that the strain data contains a glitch. Methods to incorporate iDQ into a different pipeline used to search for gravitational waves from compact binaries, GstLAL [11], were recently developed [43]. This method directly used the iDQ likelihood as an additional term in the ranking statistic of the GstLAL. Additional details on GstLAL can be found in [42, 44, 45]. Comparisons between the methods introduced in this work and those currently implemented in GstLAL are discussed in Section II C.

As use of curated data quality products has been consistently shown to increase the sensitivity of searches for gravitational waves [26, 29, 43], it is also likely that the auxiliary data used to generate these products can also benefit gravitational-wave searches. Furthermore, it is possible some useful information from the auxiliary data is discarded when curated data products are generated. Use of the raw auxiliary data is also attractive as it would reduce the time to complete an end-to-end analysis of LIGO data by no longer requiring time to generate data quality products. However, the large number of different data streams with disparate properties has made it difficult to develop generic methods to incorporate this data.

II. IMPROVED NOISE MODEL

The changing state of the detectors means that the PyCBC background will also change with respect to time. Therefore a more complete description of the noise model that accounts for this time-dependence should be given as

$$r_N(\hat{\rho};\vec{\theta};t) = \mu(\vec{\theta};t)\alpha(\vec{\theta};t)\exp\left[-\alpha(\vec{\theta};t)(\hat{\rho}-\hat{\rho}_{th})\right], \quad (6)$$

where $\mu(\vec{\theta};t)$ is the trigger density for a given template with respect to time and $\alpha(\vec{\theta};t)$ is the exponential decay rate of the background for a given template with respect to time.

Due to practical limitations, we only consider the time-dependence of the trigger density, μ, and ignore the time-dependence of the decay rate, α. The time-dependent variations that we hope to capture with this improved noise model can occur over timescales of seconds. Although a large number of triggers are identified by PyCBC per analysis, the average number of triggers per second in recent analyses is only 10 triggers per second, even though hundreds of thousands of templates are used to search the data. The modest rate of triggers, combined with the large number of templates considered, means that there is much less than 1 trigger per template per

second. This is not a sufficient number of triggers to accurately measure second-scale variations in the trigger rate. However, if we suitably bin the times and templates, this is a sufficient number of triggers to measure the time-dependence of the trigger rate. When using bins, we approximate the overall time-dependence of the trigger rate as the originally measured trigger rate multiplied by an additional bin-dependent factor.

With these approximations, the only change to the noise model is to $\mu(\vec{\theta};t)$, where the time-dependence is modeled as the product of a time-independent term, $\mu(\vec{\theta})$ and a time-dependent term $\delta(\vec{\theta};t)$. Hence the new expression for trigger density is

$$\mu(\vec{\theta};t) \approx \mu(\vec{\theta})\delta(\vec{\theta};t) \tag{7}$$

for a given θ, and time, t. The calculation of $\delta(\theta;t)$ is different depending on the type of data quality stream that is being considered.

This method makes no assumptions about the input data that is used as part of the noise model. In the case that the auxiliary data is non-informative (i.e. not correlated with times of high trigger density), the method should identify that no excess in triggers is measured, and no change to the noise model will be applied. Therefore, the sensitivity of the PyCBC search will not be impacted if non-informative auxiliary data is used by this method.

A. Binning the parameter space

In the current `PyCBC` noise model, the trigger rate, $\mu(\vec{\theta})$, is calculated for each individual template. In practice, it is not possible to also determine the time-dependent correction to the trigger rate, $\delta(\vec{\theta};t)$, for each individual template. This is due to the relatively low rate of triggers per template per second. In order to approximate the value of $\delta(\vec{\theta};t)$, we choose to group templates with similar duration into bins (denoted by θ_b). We also group times based on the value of the data quality stream, $\Omega(t)$, using additional bins (denoted by Ω_d). We then calculate a single value of $\delta(\theta_b;\Omega_d)$ for each combination of $\{\theta_b,\Omega_d\}$. These data quality bins span the range of values that the data quality stream can take and a single time-dependent correction is calculated for all times in each data quality bin. If we have N template bins and M data-quality bins, this means we only need to estimate $N \times M$ different corrections to the trigger rate. An example of how these bins could be constructed with 3 template bins and 4 data quality bins, along with the parameters of an example trigger, is shown in Figure 1.

We will label each template bin as θ_b and each data quality bin Ω_d. This means that the time-dependent correction the trigger rate in our noise model is defined as

$$\mu(\vec{\theta};t) \approx \mu(\vec{\theta})\delta(\theta_b;\Omega_d)\ . \tag{8}$$

Care must be taken when deciding on the number of bins to use in an analysis. As the presence of a gravitational-wave signal will naturally cause more triggers to be observed, there is a risk that real signals will be down-ranked if the total number of triggers produced by a signal is a significant fraction of the total number of triggers in a single bin. Conversely, if not enough bins are used, variations in time and across the template bank may not be captured. We found that having at least 50 triggers in each bin was sufficient to minimize the chance that a real signal would be artificially down-ranked.

In this work, we chose bin sizes such that the smallest bin contained at least this minimum number of triggers. This resulted in the choice of 10 template bins and either 2 (the binary case where one bin is much smaller than the other) or 200 (the non-binary case where all bins are the same size) data quality bins. This means that either 20 or 2000 different values of $\delta(\theta_b;\Omega_d)$ must be calculated for every data stream.

We construct our template bins based on template duration, with the goal of recording an equal number of triggers in each template bin. A representative chunk of LIGO data from O3 was used to calculate the specific values of the bin edges used. The bin edges are linearly spaced at $\{0,10,20\ldots,100\}$ percentile of the trigger template duration. After calculating the bin edges for this representative chunk of data, the same values of template duration were used as bin edges in all analyses. This is the default binning strategy used in this work. Two alternate binning strategies were also investigated, but were found to result in a smaller sensitivity increase than our default strategy. More details are given in Section III C.

B. Binary data quality streams

The simplest case we can consider is a binary data quality stream that only consists of 1s and 0s. Times where the data quality stream is 1 are often referred to as "active" times, and times that the stream is 0 are referred to as "inactive" times. Data quality flags are one example of a binary data stream. In this scenario, the time dependence of $\delta(\theta_b;\Omega_d)$ is also binary. We only have two data quality bins, labeled Ω_1 and Ω_0. For times that the stream is active, the time-dependent term of the noise model, $\delta(\theta_b;\Omega_1)$, is defined as

$$\delta(\theta_b;\Omega_1) = \frac{N_{b,1}}{T_1}\frac{T_{tot}}{N_{tot}} \tag{9}$$

for a given template bin, θ_b. $N_{b,1}$ is the total number of triggers in template bin b during times the data quality stream is active, while N_{tot} is the total number of triggers in the analysis. Similarly, T_1 is the total amount of time the data quality stream is active, while T_{tot} is the total amount of time in the analysis.

If the binary data quality stream is correlated with periods of high trigger density, then $\delta(\theta_b;\Omega_1) > 1$. However,

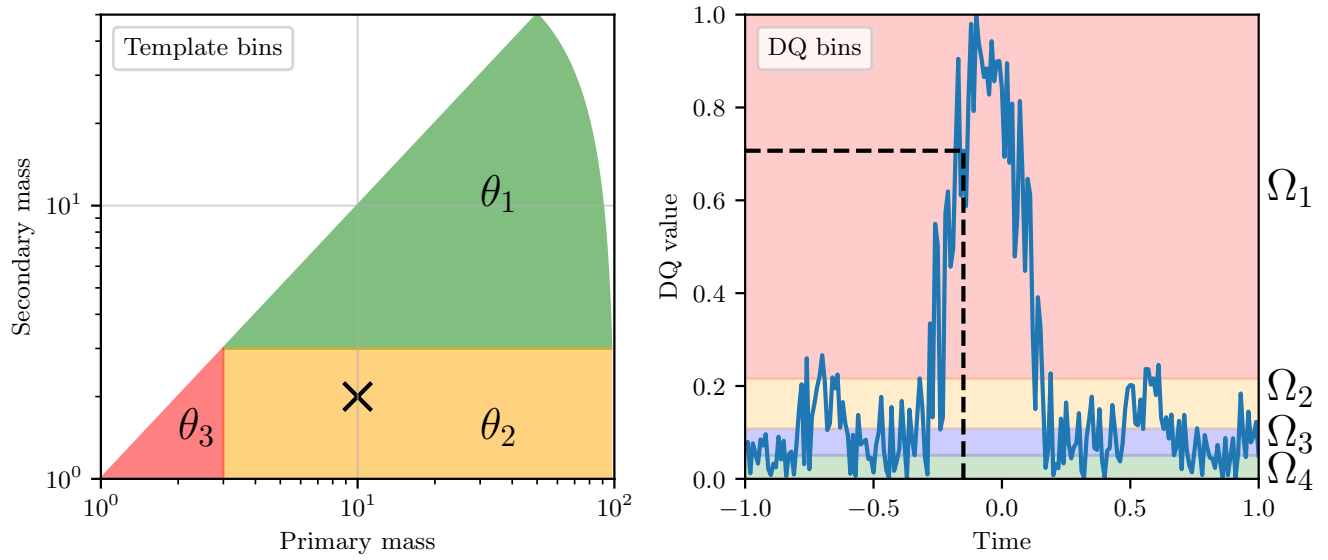

FIG. 1. An example of how the template and data quality (DQ) bins are constructed and applied. *Left:* A plot of the three different template bins corresponding to different parts of the template bank used in the search. In this case, the range of templates is characterized using the masses of the primary and secondary components of the simulated compact binary system template . An "×" marks the template parameters of an example candidate is in the template bin θ_2. *Right:* A plot of an example data stream that is used to construct four different data quality bins. The dotted line marks the time of an example candidate. In this case, the example candidates is in data quality bin Ω_1. Therefore the time-dependent term used in the `PyCBC` noise model for this candidate would be $\delta(\theta_2; \Omega_1)$.

this is not guaranteed to be the case. If $\delta(\theta_b; \Omega_1) \leq 1$, we impose $\delta(\theta_b; \Omega_1) = 1$. This is so that the data stream does not increase the significance of a candidate. All times when the data quality stream is inactive are also fixed to $\delta(\theta_b; \Omega_0) = 1$.

Directly calculating the value of $\delta(\theta_b; \Omega_1)$

When calculating this time-dependent term of the noise model, we use the merger time of each candidate and do not consider the different durations of signals across the template bank. For candidates with shorter, sub-second templates, the effect of this assumption is likely minimal, as this timescale is similar to the duration of many glitches. This assumption is less valid for longer signals as it does not account for data quality issues that may be many seconds before the time of merger of a candidate, but still overlapping the candidate. However, the signal consistency tests used by PyCBC are conversely most effective for low-mass candidates [10, 16] and have been shown to effectively mitigate the impact of data quality issues on the sensitivity of the search for such long-duration candidates. Therefore, despite this assumption, we still expect this method to increase the sensitivity of the PyCBC search in regions of the parameter space that are known to be limited by data quality issues.

C. Non-binary data quality streams

We can also consider a data quality stream that takes an arbitrarily large number of values. Such data quality streams include the `iDQ` time series or auxiliary data. In this case, we bin the data points into multiple data quality bins based on the value of each data point. The total number of bins used with this method must be tuned for each analysis. In this work, we choose to use 200 data quality bins so that each bin contained a sufficient number of triggers to reduce the bias of individual astrophysical signals.

The correction, $\delta(\theta_b; \Omega_d)$, for each data quality bin, Ω_d is calculated using the same formula as the binary case. Again similar to the binary case, we fix $\delta(\theta_b; \Omega_d) \geq 1$. Times where the data quality stream is not defined are still still used to calculate the total time and total number of triggers, but these triggers during these times are not reranked using this method.

We can also compare this correction to the model suggested for use with the `iDQ` time series in [43]. There are two main differences between our model and the model from [43]. Firstly, in our model the correction to the total trigger density is directly computed for each combination of template bin and data quality bin. This ensures that an accurate correction is applied for any data quality stream. Compared to the analytic model designed for use with the `iDQ` time series described in [43], there is a reduced risk of reranking candidates by too much or too

little. Our method also does not impose a maximum correction to the trigger density as was done in [43]. While this does introduce a risk of an arbitrarily high correction being applied, such a case would not occur unless there was indeed a strong correlation between the data quality stream and the PyCBC triggers, implying that the trigger is unlikely to come from an astrophysical signal.

D. Multiple data quality streams

For the time-dependent correction associated with two different data quality streams, δ_n and δ_m, we define the joint time-dependent correction, δ_{nm}, as

$$\delta_{nm}(\theta_b;\Omega_d) = \max\left(\delta_n(\theta_b;\Omega_d), \delta_m(\theta_b;\Omega_d)\right) . \quad (10)$$

This conservative choice ensures that if more beneficial data quality information is available, the relevant triggers will be down-ranked by a larger amount. The choice to downrank candidates by the largest time-dependent correction may lead to some triggers being down-ranked more or less than would be optimal. For example, this choice ignores any correlations between the two data quality streams. However, as both astrophysical signals and triggers caused by noise are down-ranked the same amount, we do not expect this to decrease the sensitivity of the search as compared to not using any data quality streams.

III. APPLICATIONS

One of the significant benefits of this method of incorporating data quality streams into the PyCBC search is its versatility in a variety of applications. In this section, we will demonstrate a number of use cases for this method and investigate how incorporating each data quality stream increases the detection rate of gravitational-wave signals by the PyCBC search. In all cases, we find evidence that incorporating these data quality streams can increase the number of detectable gravitational waves.

The O3 strain data used in this section from both the LIGO Hanford and LIGO Livingston detectors is available from the Gravitational Wave Open Science Center (GWOSC) [46]. Although most auxiliary data recorded by LIGO is not yet publicly available, there has been a release of auxiliary data around one event [47] and a small number of data quality products that are released publicly alongside the strain data. The majority of these analyses in this section demonstrate how data quality products not yet publicly released could be used to improve the sensitivity of the PyCBC search. The source of each data quality stream, either public or not public, is described in the relevant section.

A. Search configuration

The analyses presented in this section use data from 5 different analysis periods. These time periods correspond to the chunks of data analysed by the LIGO-Virgo collaborations during O3. The start and end times of each chunk are listed in Table I. We label each chunk by a number between 1 and 5. These chunks were chosen due to known data quality issues that may impact the sensitivity of the PyCBC search.

In all examples presented here, we use the ranking statistic introduced in this work and available as part of the PyCBC code repository found at [15]. We use a single-detector ranking statistic that includes the chi-squared test [10], the sine-Gaussian test [36], and accounts for variation in the detector's power spectral density with time [37]. We use the same template bank as was used in PyCBC analyses presented in GWTC-3 [6, 48–50]. Unless explicitly stated, triggers with a single-detector statistic above 6.5 are used to calculate the time-dependent correction to the PyCBC noise model. This threshold was tuned by hand to balance including a sufficient number of triggers to model the time-dependence and focusing on the tail of the non-Guassian distribution of triggers.

We compare the sensitivity of the search with and without incorporating data quality information by comparing the volume-time (VT) of the search in each case. This is done using a large number of simulated signals that are recovered by the PyCBC search pipeline. The distance at which a simulated signal can be detected is then used to estimate the sensitive volume. This volume multiplied by the duration of the analysis is the VT of the search. When evaluating the ratio of the VT in each analysis, we calculate the ratio of the VT by using multiple thresholds of the inverse false alarm rate (IFAR) that is assigned to each simulated signal in order to determine if a signal was detected by the search. We further present results for simulated signals with different chirp masses [51], $\mathcal{M} = (m_1 m_2)^{3/5}/(m_1+m_2)^{1/5}$ for signals produced by the merger of objects with masses m_1 and m_2. Errors for the ratio of the sensitive VT between analyses with and without DQ are estimated by calculating the VT ratio for 20 additional thresholds close to the chosen IFAR threshold.

B. iDQ time series

We analyze each of the chunks discussed above with PyCBC using the iDQ log-likelihood time series produced in low-latency as a non-binary data quality stream. The low-latency log-likelihood time series were produced by iDQ using the OVL classifier [52, 53]. This classifier was trained using triggers from 844 LIGO auxiliary data streams at each detector. This set includes all data streams that were determined to not be sensitive to gravitational-wave signals by LIGO detector characterization studies [32]. Separate instances of the classifier

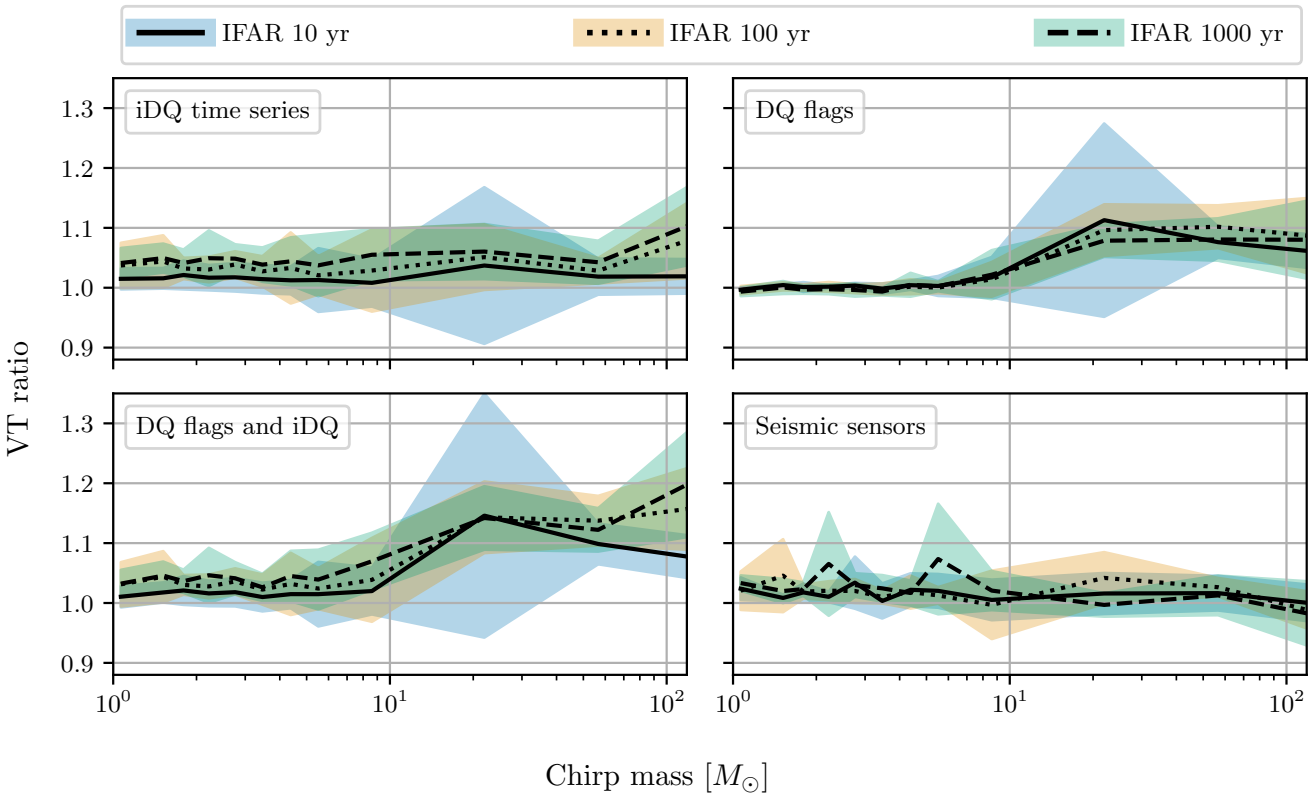

FIG. 2. The ratio of the sensitive volume-time (VT) for the PyCBC search, comparing the sensitivity of the search when using data quality (DQ) products as part of the ranking statistic versus using no data quality products. Each panel corresponds to using a different combination of data quality products. All quoted VT ratios are relative to the same PyCBC analysis that does not use any data quality products. Three different detection thresholds are considered for each combination of data quality products in addition to a range of masses of simulated signals. Shaded regions correspond to the 1σ error in the measured VT ratio for each case. The measured VT ratio for all combinations of data quality products is above 1.0, indicating that the use of these data quality products only has a positive effect. Using both DQ flags and iDQ yields the largest increase in VT of the 4 cases considered.

were trained for each interferometer used. Each instance of the classifier was trained on triggers generated from 14 days of detector data and used to make predictions until being replaced by a newly trained classifier. After the training of each classifier completed, training of a new classifier began on the most recent 14 days of data. We used the time series produced in low-latency, as opposed to time series produced offline at higher latency (and available at [54]), because we found that using the low-latency version of the iDQ timeseries led to larger increases in the sensitivity of PyCBC. This is likely due to the fact that the low-latency data was produced without the use of multiple time-chunks that were analyzed independently (as was done when producing the offline data) and was therefore more consistently normalized over the time period considered in this work [33].

Before using the iDQ time series in our analysis, we first pre-process the data stream. We downsample the log-likelihood time series from 128 Hz to 1 Hz. This is done by maximizing the iDQ time series over each integer second of data. The downsampled log-likelihood time series is then converted to percentiles, and each trigger is associated with the log-likelihood percentile at the time of the trigger. Each template bin is divided into 200 sub-bins by the triggers' iDQ log-likelihood percentiles, as described in Section II C.

We find that including the iDQ time series increases the sensitive VT of the search across the entire parameter space. This increase in search sensitivity from using the iDQ log-likelihood time series in PyCBC is shown in the upper left plot of Figure 2. We find that the gain in sensitivity generally independent of chirp mass, and is larger for higher choices of IFAR. For triggers with chirp mass above $80M_\odot$, we find a 10% increase in sensitive VT at an IFAR of 1000 years. This is the largest increase among the chirp masses and IFAR thresholds we considered.

Compared to the results of the Godwin et al. implementation of iDQ into GstLAL [43], our results show a larger increase in sensitive VT for the highest mass triggers. This is likely because we directly compute the time-dependent correction to the trigger rate instead of assuming an analytic formula for down-ranking triggers. We find that the required correction during times corre-

sponding to the highest percentiles of the iDQ time series is lower than used in the Godwin et al. implementation. It also does not down-rank any excess noise correlated with iDQ time series percentiles below 50, even in cases where we include a correction.

C. Alternate Binning Strategies

In addition to the default method of binning the template parameter space that is explained in Section II A, we investigate two alternate methods of binning the template parameter space.

In the first alternate binning method, we construct six template bins with the bin edges in a geometric series between 0.15 seconds and 150 seconds. Thus the lowest bin edge is at 0.15 seconds, and each successive bin edge is larger by a factor of $\sqrt{10}$. This method contains fewer total bins than the default method and the bins contain vastly differing numbers of triggers. However, the bin with the fewest total triggers was designed to contain roughly the same number of triggers as each of the bins in the default method.

For the second alternate binning method, we first convert the trigger template durations into percentiles. We then construct five bins with bin edges placed at percentiles of $\{0, 6.25, 12.5, 25, 50, 100\}$ in the trigger template duration based on the entire bank of templates. Similar to the first alternate method, the number of triggers in each bin is not the same, but the smallest bin is roughly the same size as each of the 10 bins when using the default method. The only difference between this binning method and the default method is the location of the bin edges; this method places the bin edges in a geometric series with different amounts of triggers in each bin while the default method uses a linear series so that each bin contains the same number of triggers.

As part of evaluating which binning method to use in this work, we compared the sensitive VT for each binning method when using the iDQ time series to analyze chunk 2. The relative VT increases from using the default binning method as compared to each of the two alternative binning strategies are shown in Figure 3. These alternate binning strategies did not increase the sensitive VT as much as the default binning method did, so they were not used in any of our other analyses.

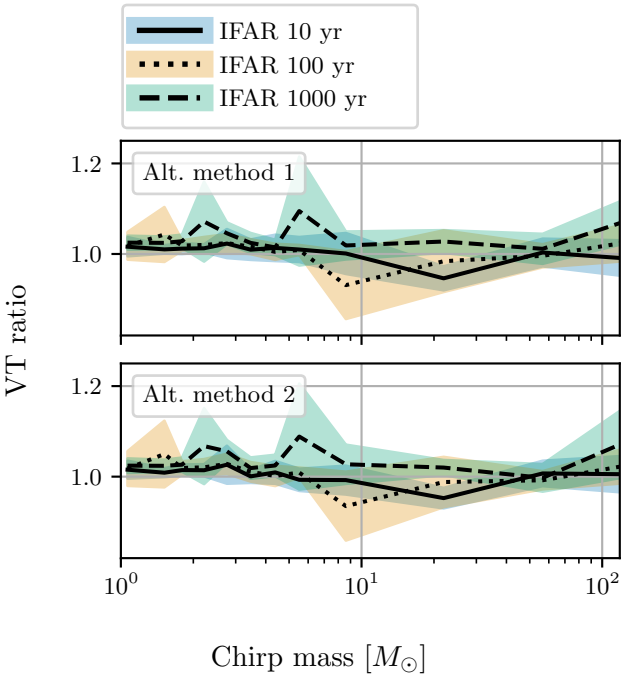

FIG. 3. The ratio of the sensitive volume-time (VT) for the PyCBC search when using the iDQ time series with different binning methods. In each case, the ratio of the measured VT when using the default binning method versus an alternate binning method is plotted. *Top*: The ratio of the sensitive VT when using the default method of binning versus an alternate method that contains bins of different sizes. *Bottom*: The ratio of the sensitive VT when using the default method of binning versus an alternate method that contains bins chosen based on the numerical value of the template durations. These alternate binning strategies perform very similarly, but the default method outperforms both alternate binning methods.

D. Data quality flags

We next investigate the benefits of using data quality flags as a part of our time-dependent noise model. The LIGO and Virgo collaboration uses a wide variety of data quality flags to indicate periods when environmental or instrumental noise sources are likely to affect the quality of the strain data. Currently, the PyCBC search uses these flags to remove triggers during these time periods from the analysis. However this reduces the analyzable time and could cause the search to miss some gravitational-wave signals. We can instead use these data quality flags as binary data streams to take into account the expected increase in the trigger rate and re-weight the detection statistic of triggers accordingly. Table I details the data quality flags active during the analysis periods that we chose to analyze. These data quality flags are released via GWOSC as a single, combined data stream [46] but are not currently publicly available separately. We chose to consider these data quality flags as multiple data streams, as each data quality flags was designed to target a different noise source, making it easier to measure the effect of these noise sources on the PyCBC trigger rate.

We also choose to calculate the time-independent portion of the PyCBC noise model after removing candidates that are present during the data quality flag segments. These candidates are still considered potential astrophysical candidates and their significance is estimated as described in section II B. We find that excluding these time periods when calculating the time-independent terms in the PyCBC noise model increases the sensitivity as compared to including them.

Chunk	GPS interval	Data Quality Flag	Flag Time	Description
1	1239641067 - 1240334090	L1:DCH-PEM_EY_ACC_BEAMTUBE_OMICRON_GT_100	1.06%	70 Hz periodic glitches due to an automated camera shutter in the End-Y station at LIGO Livingston.
2	1241724868 - 1242485150	H1:DCH-EARTHQUAKE_CS_Z_BLRMS_GT_1000	0.48%	Non-stationary noise due to high ground motion at LIGO Hanford.
		L1:DCH-THUNDER_MIC_BP_GT_300	0.05%	Excess noise due to thunderstorm at LIGO Livingston.
3	1262192836 - 1262946499	L1:DCH-WHISTLES	0.45%	Glitches caused by radio frequency (RF) beat notes at LIGO Livingston.
4	1263751734 - 1264528232	L1:DCH-WHITENED_RF45_AM_CTRL_GT_1P75	0.27%	Glitches due to 45 MHz control signal at LIGO Livingston.
		L1:DCH-WHISTLES	0.83%	Same as chunk 3.
5	1264528056 - 1265133171	L1:DCH-WHITENED_RF45_AM_CTRL_GT_1P75	0.54%	Same as chunk 4.
		L1:DCH-WHISTLES	0.16%	Same as chunk 3.

TABLE I. A list of the time periods analyzed and the data quality flags used as data quality streams in this work. All data quality flag names are sourced from [55]. Flag time refers to the analyzable time impacted by each individual flag in the analyses period.

The upper right panel of Figure 2 shows that including data quality information in the PyCBC search increases its sensitivity, in particular for high mass binaries. In this region of the parameter space, the number of detectable gravitational-wave signals increase by 10%. However, the sensitivity gains vary greatly between analysis periods. As shown in Figure 4, including data quality information in the search of chunk 1 data increases the sensitivity to signals from binary black hole mergers up to 15%. On the other hand, our approach has just a small effect for chunk 3. In fact, this period is dominated by glitches that are effectively identified and down-ranked by the PyCBC consistency tests [36].

Figure 4 also shows how our approach compares to the previous method that PyCBC use to incorporate data quality flags, namely using the data quality flag segments to veto candidates. The improvements in sensitivity are due to the increased analyzable time. This increase in sensitivity compared to using data quality flags as vetoes directly translates into more events that can be detected by PyCBC. Although the amount of time vetoed by data quality flags in recent observing runs is less than 1% [32], the high rate of detections makes it likely that some events would be missed or recovered with less significance by chance due to vetoes.

One such event, GW200129_065458, was identified by the "PyCBC-Broad" search in GWTC-3 as a coincident signal between LIGO Hanford and Virgo [6]. This event was not identified as a three-detector coincidence because the related trigger at LIGO Livingston was vetoed by a data quality flag. We find that using data quality flags for reranking triggers instead of vetoing them allows this event to also be identified at LIGO Livingston with high significance.

E. Multiple data quality products

In addition to considering the use of the iDQ time series and data quality flags separately, we investigated the benefit of using both types of data quality products at once. In cases when a trigger is down-ranked by both the iDQ time series and a data quality flag, only the larger amount of down-ranking was used, as described in section II D. We also include data quality flag information in the same way as in the previous section; triggers during data quality flags are removed when the time-independent noise model is calculated but included when candidates are identified.

We find that including both the iDQ time series and data quality flags increases the sensitivity of PyCBC by up to 20% compared to no use of data quality products, as shown in the lower left panel of Figure 2. This is roughly in line with what would be expected from adding the sensitivity increases from the individual data quality flag and iDQ results. Although the auxiliary data streams that were used to create the considered data quality flags are also used by iDQ, this result suggests that the data quality issues identified by each product are distinct.

F. Seismic monitors

Seismic activity is a major source of noise for LIGO [56–58]. Seismic noise can couple into the detector and appear as scattered light glitches [59, 60]. We use

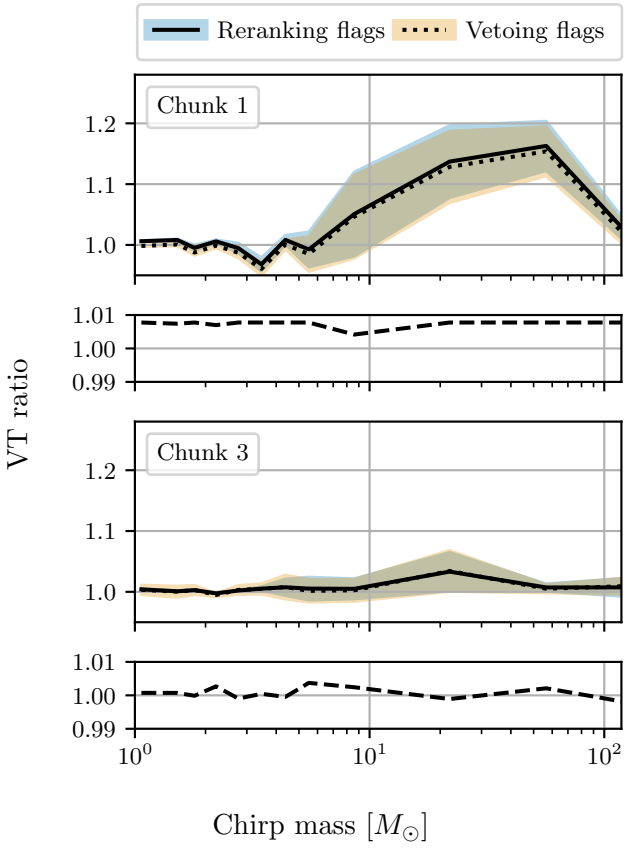

FIG. 4. The ratio of the sensitive volume-time (VT) for the PyCBC search when using data quality (DQ) flags to rerank PyCBC candidates (blue) or vetoing candidates (orange). The sensitivity is calculated at fixed inverse false alarm rate of 10 years. *Top*: Increase in search sensitivity for chunk 1, an analysis where a data quality flag was known to have a positive effect. The ratio of the search sensitivity when using reranking versus vetoing candidates is shown in the second panel. *Bottom*: Increase in search sensitivity for chunk 3, an analysis where a data quality flag was known to have minimal effect. The ratio of the search sensitivity when using reranking versus vetoing candidates is shown in the fourth panel. In both cases, reranking times during data quality flags only increases the sensitivity of the search compared to vetoing.

seismic trend data as another example of a non-binary data quality stream.

For the input data quality stream for our analysis, we use accelerometer data from the corner station at each observatory. These monitors measure ground motion in the direction perpendicular to the arms of the interferometer. The chosen data streams are focused on ground motion from 0.03 – 0.1 Hz, which is often referred to as the "earthquake band" as earthquakes are the main contributor to ground motion at these frequencies. These specific accelerometers have been chosen as they have been previously shown to be correlated with excess noise in the gravitational-wave detector data [55, 61]. Additional monitors of ground motion in directions parallel to the arms of the interferometer are located at each observatory.

For this investigation, we choose a single analysis period, chunk 2, covering from 5 May 2019 to 21 May 2019. Similar to the previous investigations, this time was chosen due to the known presence of a data quality issue that could be correlated with this data stream. This seismic data is not available for public use via GWOSC, but is displayed on the public "Detector Status" pages [62].

We found that this increased the sensitivity of the search by as much as 5% in some regions of the trigger parameter space. The increase in sensitivity from using these seismic sensors across different template masses is shown in the lower right panel of Figure 2. For most of the parameter space, only a marginal increase in sensitivity is measured. Incorporating additional sensor data may further increase these sensitivity gains. As the methods presented in this work are fully generic for any time series, any useful auxiliary information can be further incorporated into the search.

G. Large numbers of auxiliary monitors

In each observation run, hundreds of thousands of auxiliary data streams are recorded for the full duration of the run and could potentially be incorporated into the PyCBC search using the methods described in this work. However, at the time of publication, the LIGO Scientific Collaboration has only publicly released auxiliary data streams for a single data segment for a small subset of streams. This data release, containing data from 1169 data streams for 3 hours around GW170814 [63] is available at [47]. Although this amount of data is not sufficient to test if these data streams can be used to increase the sensitivity of the PyCBC search, we use this data release to demonstrate how this method can be applied for a large number of separate auxiliary monitors. We choose to only include an auxiliary data stream if a data stream with the same name was available from both sites. This reduced the total number of data streams used to 1126.

For this investigation, we made multiple changes to the standard workflow to both increase the likelihood that relevant features of included auxiliary data streams are identified as correlated with the PyCBC trigger rate and decrease the computational cost. When possible, the auxiliary data was bandpassed and the root-mean-square (RMS) of the data was calculated with a 1 second stride. The band-limited RMS is a common tool used in data quality investigations to identify time periods with excess noise [55, 61]. Furthermore, the environmental sources of noise in gravitational-wave detectors are generally most prominent at lower frequencies [28], so bandpassing the data removes the less useful high-frequency data. The targeted frequency range for this study was frequencies less than 100 Hz; specific frequency boundaries for the bandpassing were chosen based on the sample rate of the relevant data in order to restrict the data

to lower frequencies while still retaining some useful information. Data streams with a sample rate of higher than 100 Hz were bandpassed between 10 Hz and 100 Hz while streams with sample rates between 10 and 100 Hz were bandpassed between 1 Hz and 10 Hz. Data streams with sample rates below 10 Hz were instead set to the maximum value of that stream in each 1 second stride. As this investigation was not used to estimate the sensitivity of the search, we used a small template bank targeting chirp masses between $10M_\odot$ and $40M_\odot$. Due to this smaller template bank and the small amount of data considered, we lowered the SNR threshold used to calculate the trigger rate to 4.5 in order to increase the number of triggers considered.

Due to the large number of auxiliary data streams considered, it is highly likely that some sources of noise will be observed by multiple streams. In this case, the down-ranking applied is given by the description in Section II D, namely that the maximum measured down-ranking among all data streams will be applied.

The measured correlation between trigger rate in PyCBC and the 1126 data streams is shown in Figure 5. If the data streams were uncorrelated with the rate of triggers, we would expect that the distribution of the measured trigger rate in each bin would follow a Poisson distribution. A fit of the data with this distribution is shown in Figure 5 as a black dotted line. For most of these auxiliary data streams, there is no clear correlation observed between the data stream and the rate of triggers in PyCBC. However, for 10 data streams at LIGO Livingston, there is at least one data quality bin with a measured relative trigger rate above 3.25. This threshold is much higher than expected due to chance based on the fitted Poisson distribution.

The data streams that show the strongest correlation with the PyCBC trigger rate includes sensors designed to detect ground motion and magnetic noise at LIGO Livingston. Monitors of ground motion [57, 60] and magnetic noise [64] are known to be correlated with glitches in LIGO data, so it is not surprising that these data streams are the most significant outliers in the small amount data considered in this investigation. Details about these 10 outliers are included in Table II.

As auxiliary data is only publicly available for 3 hours, we were not able to use this improved noise model to reanalyze the full LIGO data set and identify new gravitational-wave candidates. However, if auxiliary data does become available, this method would allow this data to be directly used in searches for gravitational waves.

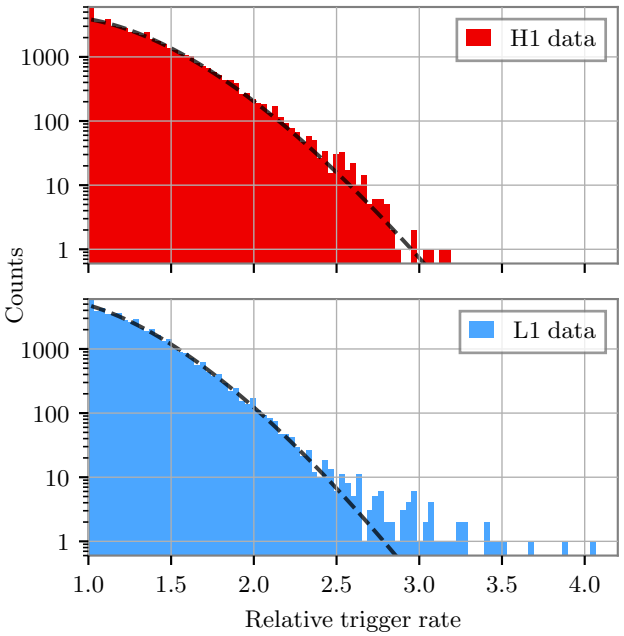

FIG. 5. Histograms of the measured trigger rate in each data quality bin from the 1126 auxiliary data streams considered in this analysis. The relative trigger rate is the ratio of the rate of triggers in each data quality bin versus the average rate of triggers at each detector. The data is fit to a Poisson distribution, shown as a black dotted line. Data from LIGO Hanford (top) shows no clear outliers, while data from LIGO Livingston (bottom) includes a small numbers of outliers based on the fitted distribution.

IV. CONCLUSIONS

We have demonstrated a novel method of directly using auxiliary data in a search for gravitational waves. This method can be applied to both the original auxiliary data and derived data quality products that are distributed alongside the strain data. Although this method was applied to the PyCBC search for compact binaries, similar methods can be incorporated to other search algorithms for both compact binaries [12, 13, 42, 65] and other gravitational-wave sources [66, 67].

With currently available data quality products, this method was able to increase the sensitivity of the PyCBC search across a wide range of masses. We find that the number of detectable gravitational-wave events is increased by up to 20% for a subset of the gravitational-wave signal population when using a combination of data products. In general, the increase in sensitivity when using these data quality streams was higher when using stricter thresholds for detection and when considering signals with higher masses. Using data quality flags as part of the PyCBC search statistic rather than to reject candidates increases the search sensitivity to 10% for the highest masses and the strictest detection threshold considered in this work. Including iDQ information via this method also increases the sensitivity by a further 5%. We have also considered using information from auxiliary data streams that monitor seismic noise, which can improve the sensitivity of PyCBC by up to 5%. Finally, we tested all auxiliary data streams that are currently publicly available [47] and identify 10 streams that show significant correlations. This method also removes the

Data stream name	Data stream description	Maximum trigger rate	P-value
L1:PEM-CS_ACC_LVEAFLOOR_BS_Z_DQ	LVEA Accelerometer	4.05	3.06×10^{-5}
L1:HPI-BS_BLND_L4C_RX_IN1_DQ	Pre-isolator motion in the global ifo basis	3.87	1.94×10^{-4}
L1:HPI-ITMY_BLND_L4C_RX_IN1_DQ	Pre-isolator motion in the global ifo basis	3.66	1.25×10^{-3}
L1:HPI-HAM3_BLND_L4C_RX_IN1_DQ	Pre-isolator motion in the global ifo basis	3.51	4.45×10^{-3}
L1:HPI-HAM3_BLND_L4C_VP_IN1_DQ	Pre-isolator motion in the global ifo basis	3.49	5.26×10^{-3}
L1:PEM-EX_MAG_VEA_FLOOR_Y_DQ	Magnetometer near ETMX chamber	3.43	8.62×10^{-3}
L1:HPI-ITMX_BLND_L4C_RX_IN1_DQ	Pre-isolator motion in the global ifo basis	3.42	9.35×10^{-3}
L1:ASC-INP1_P_IN1_DQ	Error signal for input beam in pitch	3.42	9.35×10^{-3}
L1:HPI-ITMX_BLND_L4C_RY_IN1_DQ	Pre-isolator motion in the global ifo basis	3.11	3.06×10^{-2}
L1:PEM-CS_ACC_IOT1_IMC_Z_DQ	LVEA Accelerometer	3.26	3.37×10^{-2}

TABLE II. List of the auxiliary data streams used in the search of 3 hours of data around GW170814 that are highly correlated with the rate of PyCBC triggers. All data streams with a maximum trigger rate of over 3.25 are listed. Descriptions of each data stream are sourced from [47]. The listed p-values are based on the Poisson distribution plotted in the lower panel of Figure 5 and are the probability of observing at least one instance of that value or higher in the considered dataset.

need for data quality products to be curated before use by PyCBC, reducing the time required to fully analyze LIGO data.

Ultimately, the benefits of this method are limited by the available data quality streams. Using data quality streams that are highly predictive of a high rate of PyCBC triggers will naturally increase the benefits of this method. However, compared to previous methods of incorporating data quality information, the method outlined in this work will not decrease the overall sensitivity if the auxiliary data stream is uninformative.

The versatility of this method will reduce the required effort of LIGO data quality experts to produce derived data quality products. Rather than using hand-tuned binary data quality flags, this method allows the PyCBC search to directly ingest the relevant auxiliary data stream. In addition, directly ingesting the auxiliary data stream may be more beneficial to the overall sensitivity of the search.

Similar methods can be applied to the low-latency version of the PyCBC search, PyCBC Live [68, 69]. One practical difference for a low-latency implementation of this method is that most auxiliary data streams are not available at the latencies required for detection. At present, only a subset of data quality flags and the iDQ time series are available at the required latency.

There are a number of areas of improvement for this method that could be explored in future works. First, we could add additional time dependence to our improved noise model. This method does not account for variance in the $\alpha(t)$ parameter, which also could impact sensitivity of the search. There is also an assumption that the auxiliary data stream does not include any time delay between the auxiliary data and the time of the PyCBC trigger. This may not be valid for low-mass signals that last many seconds or minutes, but data quality issues impacting this class of signals are already mitigated by signal consistency tests in the PyCBC analysis. Finally, this method could be improved by better addressing the case of multiple correlated input data streams. Addressing these limitations would require significant changes to the method introduced in this work and is therefore out of the scope of this current study.

This work presents a novel method that is able to directly use the large datasets produced at a gravitational-wave observatory in an astrophysical analysis. At present, this data is not publicly available. Hence, the maximum benefits of this work can only be realized by internal LIGO analyses. However, this method demonstrates one such practical use of directly using this dataset in astrophysical analyses and provides additional motivation for their curation and release.

ACKNOWLEDGMENTS

The authors thank the LIGO-Virgo-KAGRA PyCBC and Detector Characterization groups for their input and suggestions during the development of this work. We would like to thank Patrick Godwin for productive discussions on how to best utilize iDQ time series data, Tito dal Canton for comments on the code used in this study, and Gareth Cabourn Davies for their comments during internal review of this paper. DD is supported by the NSF as a part of the LIGO Laboratory. MT is supported by the NSF through grant PHY-2012159. SM is supported by a STFC studentship. LKN thanks the UKRI Future Leaders Fellowship for support through the grant MR/T01881X/1.

This material is based upon work supported by NSF's

LIGO Laboratory which is a major facility fully funded by the National Science Foundation. LIGO was constructed by the California Institute of Technology and Massachusetts Institute of Technology with funding from the National Science Foundation, and operates under cooperative agreement PHY-1764464. Advanced LIGO was built under award PHY-0823459. The authors are grateful for computational resources provided by the LIGO Laboratory and supported by National Science Foundation Grants PHY-0757058 and PHY-0823459. This work carries LIGO document number P2200078.

[1] J. Aasi *et al.* (LIGO Scientific Collaboration), Class. Quantum Grav. **32**, 074001 (2015), arXiv:1411.4547 [gr-qc].
[2] F. Acernese *et al.* (VIRGO), Class. Quant. Grav. **32**, 024001 (2015), arXiv:1408.3978 [gr-qc].
[3] B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, and et al., Physical Review Letters **116**, 061102 (2016), arXiv:1602.03837 [gr-qc].
[4] B. P. Abbott *et al.* (LIGO Scientific, Virgo), Phys. Rev. X , 031040 (2019), arXiv:1811.12907 [astro-ph.HE].
[5] R. Abbott *et al.* (LIGO Scientific, Virgo), Phys. Rev. X **11**, 021053 (2021), arXiv:2010.14527 [gr-qc].
[6] R. Abbott *et al.* (LIGO Scientific, VIRGO, KAGRA), (2021), arXiv:2111.03606 [gr-qc].
[7] R. Abbott *et al.* (LIGO Scientific, VIRGO), (2021), arXiv:2108.01045 [gr-qc].
[8] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series* (Wiley, 1949).
[9] C. Cutler *et al.*, Phys. Rev. Lett. **70**, 2984 (1993), arXiv:9208005 [astro-ph].
[10] B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. E. Creighton, Phys. Rev. D **85**, 122006 (2012), arXiv:0509116 [gr-qc].
[11] C. Messick *et al.*, Phys. Rev. D **95**, 042001 (2017), arXiv:1604.04324 [astro-ph.IM].
[12] F. Aubin *et al.*, Class. Quant. Grav. **38**, 095004 (2021), arXiv:2012.11512 [gr-qc].
[13] Q. Chu *et al.*, Phys. Rev. D **105**, 024023 (2022), arXiv:2011.06787 [gr-qc].
[14] T. Venumadhav, B. Zackay, J. Roulet, L. Dai, and M. Zaldarriaga, Phys. Rev. D **101**, 083030 (2020), arXiv:1904.07214 [astro-ph.HE].
[15] A. Nitz *et al.*, "gwastro/pycbc: Pycbc release v2.0.2 (v2.0.2)," (2022).
[16] S. A. Usman *et al.*, Class. Quant. Grav. **33**, 215004 (2016), arXiv:1508.02357 [gr-qc].
[17] A. H. Nitz, T. Dent, T. Dal Canton, S. Fairhurst, and D. A. Brown, Astrophys. J. **849**, 118 (2017), arXiv:1705.01513 [gr-qc].
[18] G. S. Davies, T. Dent, M. Tápai, I. Harry, C. McIsaac, and A. H. Nitz, Phys. Rev. D **102**, 022004 (2020), arXiv:2002.08291 [astro-ph.HE].
[19] A. H. Nitz, T. Dent, G. S. Davies, S. Kumar, C. D. Capano, I. Harry, S. Mozzon, L. Nuttall, A. Lundgren, and M. Tápai, Astrophys. J. **891**, 123 (2020), arXiv:1910.05331 [astro-ph.HE].
[20] A. H. Nitz, C. D. Capano, S. Kumar, Y.-F. Wang, S. Kastha, M. Schäfer, R. Dhurkunde, and M. Cabero, Astrophys. J. **922**, 76 (2021), arXiv:2105.09151 [astro-ph.HE].
[21] A. H. Nitz, S. Kumar, Y.-F. Wang, S. Kastha, S. Wu, M. Schäfer, R. Dhurkunde, and C. D. Capano, (2021), arXiv:2112.06878 [astro-ph.HE].
[22] A. Buonanno, B. R. Iyer, E. Ochsner, Y. Pan, and B. S. Sathyaprakash, Phys.Rev. **D80**, 084043 (2009), arXiv:0907.0700 [gr-qc].
[23] A. Bohé *et al.*, Phys. Rev. D **95**, 044028 (2017), arXiv:1611.03703 [gr-qc].
[24] B. Abbott *et al.* (LIGO Scientific Collaboration), Phys. Rev. D **72**, 082001 (2005), arXiv:gr-qc/0505041.
[25] M. Cabero *et al.*, Class. Quant. Grav. **36**, 15 (2019), arXiv:1901.05093 [physics.ins-det].
[26] D. Davis, L. V. White, and P. R. Saulson, Class. Quant. Grav. **37**, 145001 (2020), arXiv:2002.09429 [gr-qc].
[27] A. Buikema *et al.* (aLIGO), Phys. Rev. D **102**, 062003 (2020), arXiv:2008.01301 [astro-ph.IM].
[28] P. Nguyen *et al.* (AdvLIGO), Class. Quant. Grav. **38**, 145001 (2021), arXiv:2101.09935 [astro-ph.IM].
[29] B. P. Abbott *et al.* (LIGO Scientific Collaboration, Virgo Collaboration), Class. Quant. Grav. **35**, 065010 (2018), arXiv:1710.02185 [gr-qc].
[30] J. Driggers *et al.* (LIGO Scientific), Phys. Rev. D **99**, 042001 (2019), arXiv:1806.00532 [astro-ph.IM].
[31] D. Davis, T. J. Massinger, A. P. Lundgren, J. C. Driggers, A. L. Urban, and L. K. Nuttall, Class. Quant. Grav. **36**, 055011 (2019), arXiv:1809.05348 [astro-ph.IM].
[32] D. Davis *et al.* (LIGO), Class. Quant. Grav. **38**, 135014 (2021), arXiv:2101.11673 [astro-ph.IM].
[33] R. Essick, P. Godwin, C. Hanna, L. Blackburn, and E. Katsavounidis, Machine Learning: Science and Technology **2**, 015004 (2020).
[34] B. P. Abbott *et al.* (LIGO Scientific Collaboration, Virgo Collaboration), Class. Quant. Grav. **33**, 134001 (2016), arXiv:1602.03844 [gr-qc].
[35] T. Akutsu *et al.* (KAGRA), Nature Astron. **3**, 35 (2019), arXiv:1811.08079 [gr-qc].
[36] A. H. Nitz, Class. Quant. Grav. **35**, 035016 (2018), arXiv:1709.08974 [gr-qc].
[37] S. Mozzon, L. K. Nuttall, A. Lundgren, T. Dent, S. Kumar, and A. H. Nitz, Class. Quant. Grav. **37**, 215014 (2020), arXiv:2002.09407 [astro-ph.IM].
[38] B. Zackay, T. Venumadhav, J. Roulet, L. Dai, and M. Zaldarriaga, Phys. Rev. D **104**, 063034 (2021), arXiv:1908.05644 [astro-ph.IM].
[39] The Virgo Collaboration, "Virgo Detector Characterization and Data Quality during the O3 run," (2020), In preparation.
[40] R. Abbott *et al.* (LIGO Scientific, VIRGO, KAGRA), (2022), arXiv:2203.01270 [gr-qc].
[41] S. Anderson and R. Williams, *LIGO Data Management Plan: June 2017*, Tech. Rep. LIGO-M1000066-v25 (LIGO, 2021).
[42] S. Sachdev *et al.*, (2019), arXiv:1901.08580 [gr-qc].
[43] P. Godwin *et al.*, (2020), arXiv:2010.15282 [gr-qc].
[44] C. Hanna *et al.*, Phys. Rev. D **101**, 022003 (2020),

arXiv:1901.02227 [gr-qc].
[45] K. Cannon *et al.*, (2020), arXiv:2010.05082 [astro-ph.IM].
[46] R. Abbott *et al.* (LIGO Scientific, Virgo), SoftwareX **13**, 100658 (2021), arXiv:1912.11716 [gr-qc].
[47] "Auxiliary Channel Three Hour Release," https://www.gw-openscience.org/auxiliary/GW170814/.
[48] S. Roy, A. S. Sengupta, and N. Thakor, Phys. Rev. D **95**, 104045 (2017), arXiv:1702.06771 [gr-qc].
[49] S. Roy, A. S. Sengupta, and P. Ajith, Phys. Rev. D **99**, 024048 (2019), arXiv:1711.08743 [gr-qc].
[50] T. Dal Canton and I. W. Harry, (2017), arXiv:1705.01845 [gr-qc].
[51] C. Cutler and E. E. Flanagan, Phys. Rev. D **49**, 2658 (1994), arXiv:9402014 [gr-qc].
[52] R. Essick, L. Blackburn, and E. Katsavounidis, Class. Quant. Grav. **30**, 155010 (2013), arXiv:1303.7159 [astro-ph.IM].
[53] P. Godwin, *Low-latency Statistical Data Quality in the Era of Multi-Messenger Astronomy*, Ph.D. thesis, Pennsylvania State University (2020).
[54] P. Godwin *et al.*, *GWTC-2 Data Release: Renormalized iDQ time series*, Tech. Rep. DCC-T2100059 (LIGO, 2021).
[55] D. Davis *et al.*, *Data Quality Vetoes Applied to the Analysis of LIGO Data from the Third Observing Run*, Tech. Rep. DCC-T2100045 (LIGO, 2021).
[56] M. Coughlin, C. Stubbs, S. Barrientos, C. Claver, J. Harms, C. Smith, and M. Warner, Exper. Astron. **39**, 387 (2015), arXiv:1412.3603 [astro-ph.IM].
[57] D. M. MacLeod, S. Fairhurst, B. Hughey, A. P. Lundgren, L. Pekowsky, J. Rollins, and J. R. Smith, Class. Quant. Grav. **29**, 055006 (2012), arXiv:1108.0312 [gr-qc].
[58] E. Schwartz *et al.* (LIGO), Class. Quant. Grav. **37**, 235007 (2020), arXiv:2007.12847 [physics.ins-det].
[59] T. Accadia *et al.*, Class. Quant. Grav. **27**, 194011 (2010).
[60] S. Soni *et al.* (LIGO), Class. Quant. Grav. **38**, 025016 (2020), arXiv:2007.14876 [astro-ph.IM].
[61] D. Davis *et al.*, *Data Quality Vetoes Applied to the Analysis of LIGO Data from the Second Observing Run*, Tech. Rep. DCC-T2100150 (LIGO, 2021).
[62] Gravitational-wave Open Science Center, "Gravitational-wave observatory status," https://www.gw-openscience.org/detector_status/.
[63] B. P. Abbott *et al.* (LIGO Scientific Collaboration, Virgo Collaboration), Phys. Rev. Lett. **119**, 141101 (2017), arXiv:1709.09660 [gr-qc].
[64] L. K. Nuttall, Phil. Trans. Roy. Soc. Lond. **A376**, 20170286 (2018), arXiv:1804.07592 [astro-ph.IM].
[65] T. Venumadhav, B. Zackay, J. Roulet, L. Dai, and M. Zaldarriaga, Phys. Rev. D **100**, 023011 (2019), arXiv:1902.10341 [astro-ph.IM].
[66] S. Klimenko *et al.*, Phys. Rev. D **93**, 042004 (2016), arXiv:1511.05999 [gr-qc].
[67] R. Lynch, S. Vitale, R. Essick, E. Katsavounidis, and F. Robinet, Phys. Rev. D **95**, 104046 (2017), arXiv:1511.05955 [gr-qc].
[68] A. H. Nitz, T. Dal Canton, D. Davis, and S. Reyes, Phys. Rev. D **98**, 024050 (2018), arXiv:1805.11174 [gr-qc].
[69] T. Dal Canton, A. H. Nitz, B. Gadre, G. S. Cabourn Davies, V. Villa-Ortega, T. Dent, I. Harry, and L. Xiao, Astrophys. J. **923**, 254 (2021), arXiv:2008.07494 [astro-ph.HE].