

# Identification and removal of non-Gaussian noise transients for gravitational wave searches

Benjamin Steltner,<sup>1,2,a</sup> Maria Alessandra Papa,<sup>1,3,2,b</sup> and Heinz-Bernd Eggenstein<sup>1,2,c</sup>

<sup>1</sup>Max Planck Institute for Gravitational Physics (Albert Einstein Institute), Callinstrasse 38, 30167 Hannover, Germany

<sup>2</sup>Leibniz Universität Hannover, D-30167 Hannover, Germany

<sup>3</sup>University of Wisconsin Milwaukee, 3135 N Maryland Ave, Milwaukee, WI 53211, USA

(Dated: May 21, 2021)

We present a new *gating* method to remove non-Gaussian noise transients in gravitational wave data. The method does not rely on any a-priori knowledge on the amplitude or duration of the transient events. In light of the character of the newly released LIGO O3a data, glitch-identification is particularly relevant for searches using this data. Our method preserves more data than previously achieved, while obtaining the same, if not higher, noise reduction. We achieve a  $\approx 2$ -fold reduction in zeroed-out data with respect to the gates released by LIGO on the O3a data. We describe the method and characterise its performance. While developed in the context of searches for continuous signals, this method can be used to prepare gravitational wave data for any search. As the cadence of compact binary inspiral detections increases and the lower noise level of the instruments unveils new glitches, excising disturbances effectively, precisely, and in a timely manner, becomes more important. Our method does this. We release the source code associated with this new technique and the gates for the newly released O3 data.

## I. INTRODUCTION

While many loud gravitational wave signals have been detected, much of the high precision science and new discoveries in the nascent field of gravitational wave astronomy will benefit from noise-characterization and noise-mitigation techniques [1–7, 23].

The data of gravitational wave detectors is dominated by noise. This noise is by and large Gaussian with a stable spectrum, but  $\lesssim 10\%$  of it may be infested by high-powered short-lived disturbances (glitches) and by nearly monochromatic coherent spectral artefacts (lines) in a variety of amplitudes, from extremely large to extremely weak.

Typically the short-lived glitches affect the sensitivity of short-lived signal searches while the coherent lines affect the sensitivity of searches for persistent signals. But when a short-lived glitch is powerful enough, it can also temporarily degrade the sensitivity of searches for long-lived signals, by increasing the average noise-floor level in a broad frequency range for its duration.

Two noise-mitigating techniques are typically used to prepare the gravitational wave data for searches: *gating*, performed in the time domain and *line-cleaning*, performed in the frequency domain. Broadly speaking, the former is used to remove loud glitches and the latter to remove spectral lines. The latter is usually only used in searches for persistent signals or stochastic backgrounds [24, 27, 29]

In this paper we illustrate a new gating application, *gatestrain*, which enables a more precise removal of

glitches compared with other widely used and publicly available gating methods, discarding significantly less data. Furthermore our gating procedure does not rely on any single fixed threshold that establishes what data should be gated, but rather it adjusts the threshold based on the achieved noise reduction. These are important features when the glitches vary much from data-set to data-set, and within the same same data-set, because the method does not require time-intensive tuning of ad-hoc parameters.

We publish the gates found with our new method on O3a LIGO data as well as the new gating application in the supplementary material [18].

The paper is organised as follows. In Section II we describe the noise disturbances in Advanced LIGO data, which are particularly detrimental to continuous wave searches, and the typical mitigation techniques used to prepare the data before performing such searches. In Section III we present the idea of time-domain gating and explain our new method *gatestrain*. The performance of *gatestrain* on Advanced LIGO data from the first, second and third observation runs (O1, O2 and O3) is presented in Section IV. In Section V we discuss our results.

## II. NOISE AND MITIGATION TECHNIQUES

The present generation of gravitational-wave detectors operates in a noise-dominated regime. The noise is primarily Gaussian with two main types of deviations: short-lived non-Gaussian transients - *glitches* - and long-lived nearly monochromatic coherent artefacts - *lines*.

Lines are instrumental or environmental disturbances manifesting as narrow spectral artefacts, sometimes visible as lines in the frequency domain of the raw data. These disturbances lead to false candidates in continuous

<sup>a</sup> benjamin.steltner@aei.mpg.de

<sup>b</sup> maria.alessandra.papa@aei.mpg.de

<sup>c</sup> heinz-bernd.eggenstein@aei.mpg.de

wave searches and in searches for stochastic backgrounds [25, 26].

These false candidates are nefarious for multiple reasons: With limited computing power, the search sensitivity is degraded because computing power is spent evaluating false candidates rather than more promising ones. In broad parameter searches for continuous signals, only the top results can be returned by the search, and if these “top lists” are swamped by disturbances, it means that real signals, with lower scores, can never be inspected. Finally the evaluation of the significance of any finding becomes a lot harder in the presence of contaminants.

Even though the identification of false signal-candidates can happen on the actual search result [26], a standard way to deal with lines has been to replace the affected frequency bins with Gaussian noise in the data input to the search, not allowing the excess power to “spread” to many signal-frequency results. This method is called *line cleaning*. Line-cleaning relies on knowing where the spectral contamination occurs and hence on detector-characterization studies such as [16] that produce the so-called “lines lists” that LIGO releases together with its data.

The line-cleaning procedure eliminates false candidates but also turns the search “deaf” to signals’ contributions at the cleaned-out frequencies. This complicates the interpretation of the upper limits in the bands containing those frequencies. All the same, line-cleaning has been adopted in many searches and the upper limits are either not set at all in “cleaned out” bands [28] or the cleaning is folded-in the upper limit procedure resulting in a slightly degraded upper limit in those bands [29].

Loud glitches impact the sensitivity of transient signal searches by contributing to the background distribution used to estimate the significance of any finding. But they also degrade persistent-signal searches in two ways: 1) a high-power glitch directly increases the noise floor in a broad frequency range and 2) a loud glitch invalidates one of the assumptions of the line-cleaning method, and introduces artefacts in the cleaned data. We will discuss this latter point in Section IV B.

A typical mitigation technique for glitches is *gating*: the time domain data affected by a glitch is simply removed. Gating is a standard step of the compact-binary coalescence search pipeline `pyCBC` [13], but continuous wave search pipelines also use it [14, 15, 23]. In fact the O1 data Einstein@Home search for continuous waves from Cassiopeia A, Vela Jr. and G347.3 [27, 28] used the gating on its data and specifically used the `pyCBC` gating module because of its ease of use and prompt availability.

### III. TIME-DOMAIN GATING

The core idea of gating is to detect and remove glitches in the time domain. The different applications differ mostly in how the glitch detection is done and what classifies as being part of the glitch. This has implications

on the effectiveness of the gating and/or on how much “tinkering” and tuning is necessary to achieve optimal gating in any specific data set.

Typically gating in preparation for transient signal searches tends to be less aggressive in removing data than the gating in preparation for persistent signal searches, by using shorter gates and higher thresholds. With this approach a number of glitches survive, increasing the false alarm rate, but transient signals that may happen near glitches are not discarded together with the glitch. Persistent signals on the other hand are always-on, thus even with the most aggressive cleaning, only a small fraction of signal is removed.

In this Section we describe our gating method that presents two novelties with respect to publicly available gating methods: 1) the adaptive determination of the gate duration 2) the self-adjusting amplitude threshold for gate-identification, based on an iterative data-quality check of the gated data.

In most gating methods employed in gravitational wave searches, the glitch detection does not happen on the raw data. Our gating procedure uses the same initial signal-processing steps as `pyCBC`, up to the actual detection of glitches, when the two methods differ. The data is divided in chunks, ranging in duration from O(seconds) typically in compact binary coalescence searches, to O(tens of minutes) typically in continuous wave searches. In [27, 28] we used chunks that are 1 800 s long. For each chunk the following steps are taken:

- high-pass the data with an 8th-order Butterworth filter at 10 Hz. Since the released Advanced LIGO data is not to be used for astrophysical searches below 10 Hz, we refer to this as the input data.
- let  $\mathcal{P}_r(f)$  be the power spectral density noise floor (with units [1/Hz]) estimated from this data. Since the data is not stationary, in order to produce this “reference” power spectral density we divide the chunk in O(100) segments, compute the noise spectrum on each of these and, bin per bin, take the median over all the realisations.
- take the Fourier transform  $\tilde{h}(f)$ , whiten and obtain:
 
$$\tilde{h}_w(f) = \frac{\tilde{h}(f)}{\sqrt{\mathcal{P}_r(f)}}$$
- Inverse Fourier transform and obtain  $h_w(t)$  ([ $\sqrt{\text{Hz}}$ ])
- gate the  $h_w(t)$  time-series

This whitening process produces a  $h_w(t)$  time-series that in Gaussian noise has a mean  $\mu = 0$  and standard deviation  $\sigma = 1.0$  [3], with similar contributions from all frequencies. When a glitch happens, it is more visible in  $h_w(t)$  than in the original  $h(t)$ , because its contribution is not “hidden” by loud noise from the low frequencies. This is shown clearly in Figure 2.

Periods that harbor glitches are identified in  $h_w(t)$ . The data in these periods is set to zero with a Tukey

taper on either side of the period. With the expression gate,  $g_i$ , we refer to each set of neighbouring data points whose original value has been set to zero and their time-stamps:  $g_i = (\{t\}_i, \{h_w^g\}_i)$ .

The pyCBC gates are established based on two parameters: a threshold  $H$  and a duration  $\tau_{\text{dur}}$ . All times  $t_k$  are recorded where  $|h_w(t_k)| > H$ . If more points above threshold are closer than  $\tau_{\text{dur}}$  they are considered part of the same gate. The gated data is extended further to the left of the earliest and to the right of the latest point above threshold, by at most  $\tau_{\text{dur}}/2$  and based on where the highest data in the first and last  $\tau_{\text{dur}}/2$  interval is. Finally a Tukey taper is applied to either side of the gate to prevent discontinuities [13]. The typical values for transient signal searches, e.g. in the first gravitational wave transient catalogs [8–12], are  $H = 100\sqrt{\text{Hz}}$  ( $25\sqrt{\text{Hz}}$ ),  $\tau_{\text{dur}} = 0.25\text{ s}$  ( $0.125\text{ s}$ ) and a Tukey taper of  $0.25\text{ s}$  ( $0.125\text{ s}$ ) (for the latest catalogs). In [28] we used  $H = 50\sqrt{\text{Hz}}$ ,  $\tau_{\text{dur}} = 16\text{ s}$  and a Tukey taper of  $0.25\text{ s}$ .

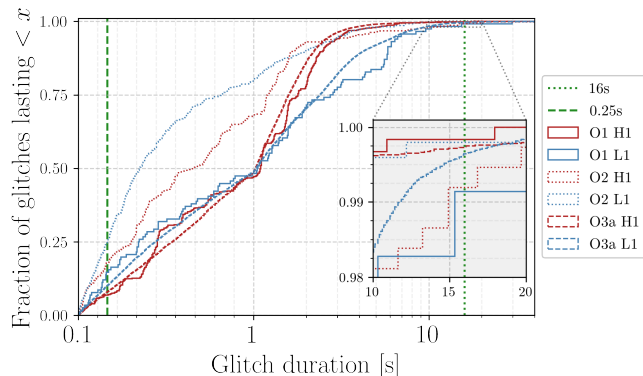


FIG. 1. Cumulative histogram of measured glitch durations. Depending on run and detector 50% to 80% of glitches last less than one second, while less than a few percent last longer than  $O(10\text{ s})$ . The maximum glitch duration is a few tens of seconds for H1 in O1 and O2 data, and L1 in O1 data, and reaches nearly 300 s in L1 O2 data. The glitch durations are measured with `gatestrain`. Note that the x-axis is displayed in symlog, i.e. linear between 0 – 1 s and logscale above.

Fig. 1 shows that glitches can last from fractions of a second to a few tens of seconds, with more than 60% of the glitches in the O2 data lasting less than 1s. It also shows that there is a great variability in glitch duration, depending on the detector and on the run. For instance, in O1  $\sim 50\%$  of glitches in either detectors last less than 1 s, whereas in O2  $\sim 50\%$  of the L1 glitches last less than 0.3s. This variability is hard to capture with simple glitch-detection schemes: for example for glitches having “long tails” that do not make it above the single threshold, those tails remain undetected and are excluded from the gates.

We develop a more generic glitch-identification and -removal scheme, with a varying gate size, estimated on the data itself. This is particularly relevant when other data, for example from environmental monitors around

the detectors, is not available, as for the gravitational wave data releases.

Our method uses three parameters: a high threshold  $H^{\text{high}}$ , a low threshold  $H^{\text{low}}$  and a duration parameter  $\tau_{\text{dur}}$ . The gates are constructed as follows:

- All times  $t_k^h$  are recorded where  $|h_w(t_k^h)| > H^{\text{high}}$ .
- All times  $t_j^\ell$  are recorded where  $|h_w(t_j^\ell)| > H^{\text{low}}$ .
- The  $t_j^\ell$  are then divided in groups such that for each member of the group there exists at least another member closer than  $\tau_{\text{dur}}$ . When there are no nearby points, a single-member group is created.
- We only keep those groups such that there exists at least a  $t_k^h$  closer than  $\tau_{\text{dur}}$  to at least one member of the group.
- For each of the surviving groups: all timestamps between the earliest and latest, plus a Tukey taper to either side, constitute a gate.

The low threshold is set as  $H^{\text{low}} = n^\ell \sigma$ , where  $\sigma$  is an estimate of the standard deviation of well-behaved parts of the data. We have used the harmonic mean of the standard deviation of  $h_w(t)$  from shorter duration chunks, say  $\approx 10\text{ s}$  long, out of the the 30 minute segment under consideration. We use the harmonic mean so that  $\sigma$  is not affected by the presence of disturbances. We set  $n^\ell$  to be high enough that Gaussian noise fluctuations at such level are rare, typically  $n^\ell \approx 5.5$ .

The high threshold is crucial because whether a glitch is identified, hinges on there being  $|h_w(t)|$  values above  $H^{\text{high}}$ . A too low  $H^{\text{high}}$  leads to too many unnecessary gates and thus wasted data, while a too high  $H^{\text{high}}$  leads to missed glitches. We use an iterative lowering of  $H^{\text{high}}$  and evaluate the performance of the gating at each threshold. We stop lowering the threshold when it has reached a pre-set minimum value or when the measured performance is satisfactory.

As an indicator of the performance of the gating we take the quantity

$$R = \frac{1}{N_f} \sum_{f_i} \frac{\mathcal{P}(f_i)}{\mathcal{P}_r(f_i)}, \quad (1)$$

where  $\mathcal{P}$  is the power spectral density from the gated data and  $\mathcal{P}_r$  is the reference power spectral density described in Section III<sup>1</sup>. The sum is over frequency bins  $f_i$ . Experience has shown that using a  $\approx 5 \sim 10\text{ Hz}$  band between 25 Hz and 40 Hz, depending on the run, without loud lines or disturbances is suitable to identify most

<sup>1</sup> In `gatestrain` it is alternatively possible to specify an external file which holds the reference PSD. This could e.g. be calculated by taking the harmonic mean over the full run. Since the detector changes on various timescales, this method is not recommended.

glitches. The reason for this lies in the character of the LIGO data, with most glitches having spectral content at lower frequencies. A value of  $R \approx 1$  indicates that the gated time series is not / no more affected by glitches. Value of  $R > 1$  indicates the presence of glitches in the gated data.

At the first iteration we use a high threshold,  $H^{\text{high}} \approx 50 \sigma$ , gate the data and compute  $R$ . If this ratio exceeds a threshold  $R^{\text{th}}$ , we reduce  $H^{\text{high}}$ , gate the data and check  $R$  again. We continue until either  $H^{\text{high}}$  reaches a minimum value or  $R$  becomes small enough. In the O1 and O2 data we found that decreasing the  $H^{\text{high}}$  threshold by 10 at each iteration, and setting the  $R$  and the  $H^{\text{high}}$  thresholds to 1.05 and 20, respectively, achieved stable and good performance. On O3 data the same choices gave very good performance. We note that a reasonable choice for the  $H^{\text{high}}$  could be to set it equal to  $H^{\text{low}}$ , but in O1 and O2 data we observe that this leads to sacrificing a lot more data, for a very small decrease in noise level. For this reason we leave it as a free parameter.

An example of this process with  $\tau_{\text{dur}} = 3$  s and a Tukey window of 0.25 s, is shown in Fig. 2 (time-domain) and Fig. 3 (frequency-domain). Three glitches are clearly seen in the time-domain plot. In the first iteration, with the highest  $H^{\text{high}}$  threshold, only the middle peak is detected and removed. The resulting amplitude spectral density (ASD, equal to  $\sqrt{\mathcal{P}}$ ) is shown in purple in Figure 3. The comparison with the reference  $\mathcal{P}_r$  yields  $R \geq 1.32$  and indicates that there could be more glitches, so the process continues with a lower values of  $H^{\text{high}}$ . In the second iteration  $H^{\text{high}} = 40$  and the second glitch is included. After removing the second glitch  $R \leq 1.05$  and this concludes the gating procedure. The third peak is thus not gated as it has not enough impact on the sensitivity. For comparison the ASD after removal of the third peak is also shown.

Data	SFTs	Data [h]	Data [d]
O1-H1	3124	1562	65.1
O1-L1	2120	1060	44.2
O2-H1	5066	2533	105.5
O2-L1	4984	2492	103.8
O3-H1	5977	2988.5	124.5
O3-L1	6377	3188.5	132.9

TABLE I. Data that we used the gating procedure on.

#### IV. RESULTS

We consider public data from the O1, O2 and O3a Advanced LIGO runs [17, 19, 20]. We produce half-hour baseline (Short) Fourier transforms, SFTs [21], as summarized in Table I. We prepare different data sets, de-

Data	Method	SFTs w/ gates	Gates	How much data zeroed-out	
				[s]	[h]
O1-H1	pyCBC	667	884	12 360.62	3.43
O1-H1	gatestrain	686	799	827.20	0.23
O1-L1	pyCBC	173	222	3 110.29	0.86
O1-L1	gatestrain	183	205	271.05	0.08
O2-H1	pyCBC	708	784	12 453.41	3.46
O2-H1	gatestrain	852	980	479.29	0.13
O2-L1	pyCBC	620	692	10 603.56	2.95
O2-L1	gatestrain	981	1 151	723.94	0.20
O3-H1	LVC	5 695	20 205	141 070.9	39.2
O3-H1	gatestrain	4 885	11 581	38 236.8	10.6
O3-L1	LVC	6 366	49 653	1 441 915.8	400.5
O3-L1	gatestrain	5 825	21 525	742 985.6	206.4

TABLE II. Total amount and duration of gates for each detector / observing run produced by `gatestrain` and `pyCBC` as used in [28] or LIGO’s procedure used on O3 data [31]. Since `pyCBC` gates may overlap, their total duration is less than the total number of gates times 16 s.

pending on the run, to compare the performance of our method with existing ones. Table II summarizes how much data was gated by the different procedures.

##### A. Gating the O1 and O2 LIGO data

We prepare three different sets, one without gating, one with the `pyCBC` gating procedure used in [28], with  $\tau_{\text{dur}}$  conservatively set to 16 s, and one with our new gating `gatestrain`.

Table II shows how many gates were used and how much time was zeroed-out by each procedure. While the number of gates of our procedure is similar or larger than the number of gates identified by `pyCBC`, overall our `gatestrain` removes much less data:  $\sim 4 - 9\%$  of what is removed by `pyCBC`. For instance of the 3124 O1-H1 SFTs, 686 are affected by one or more glitches which are gated with a total of 799 gates and 0.23 h of time domain data lost. In comparison, `pyCBC` gating results in a slightly lower number of affected SFTs (667) but 3.43 h of lost data. For the L1 detector both methods find roughly  $\sim 180$  O1 SFTs to be affected by glitches, where `pyCBC` gating removes 0.86 h of time domain data while `gatestrain` removes 0.08 h.

The noise floor of gated data is lower than that of ungated data. Fig. 3 compares the amplitude spectral density of our gated data and ungated data.

The actual improvements differ between detectors and runs: an improvement of a factor greater than 3 is seen

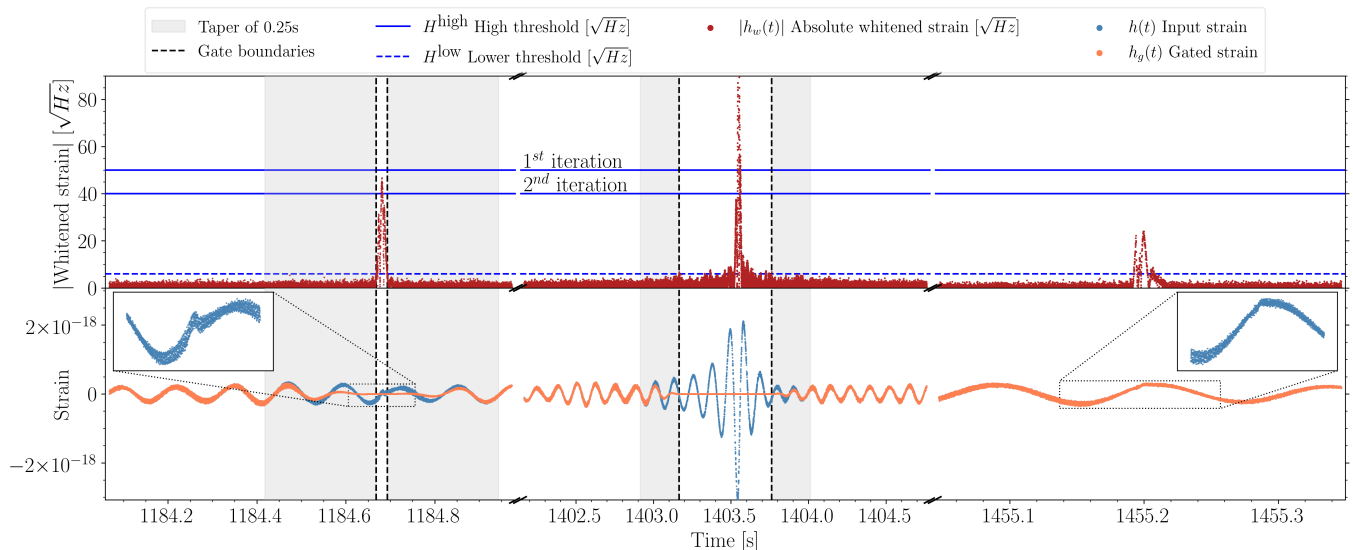


FIG. 2. Example of how the gating procedure works using three snippets of data spanning  $\approx 270$  s. The lower panel shows the original strain data  $h(t)$  (blue) and the gated  $h_g(t)$  (orange). The insets magnify the input strain to show where the glitch occurs. The three snippets of data present glitches of different size. The upper panel shows the absolute value of the whitened time series  $|h_w(t)|$  (red), which is the quantity used to detect glitches, as explained in the main text. The glitch-detection threshold are the (blue) horizontal lines, solid and dashed.

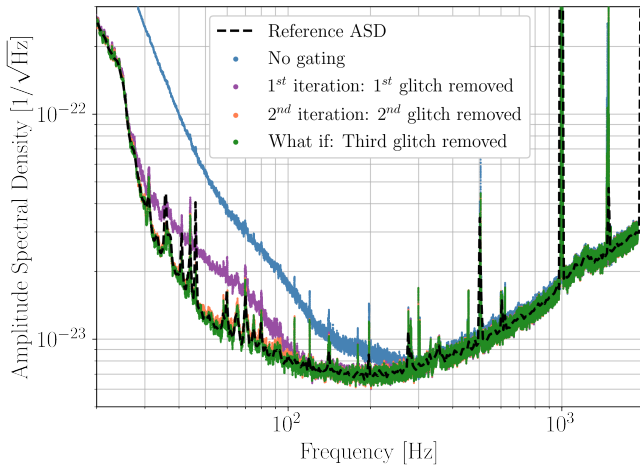


FIG. 3. Amplitude spectral density of the data during the 1800s from which the snippets of Fig. 2 are taken. We show the ASD of the data at different stages of the gating process. The noise floor of the original data (top curve, blue) is significantly higher than the reference ASD (dashed line). In the first iteration (purple) the glitch from the central data snippet is removed, resulting in a vastly improved ASD, but the comparison to the reference ASD shows potential for further improvement below  $\sim 100$  Hz. In the second iteration the left data-snippet glitch of the previous figure is found and removed, and this lowers the noise floor the low-frequency region (orange). At this point gating ends because the reference power spectral density  $\mathcal{P}_r$  is matched. The glitch in the right hand-side data snippet is not large enough to make a difference and it is left ungated. For reference, the bottom line (green) shows the ASD after removing this third glitch, which our procedure does not do.

in O1-H1 data, in the highest sensitivity region in frequency, and of  $\sim 4\%$  in O1-L1 data. In O2 data gating significantly decreases the noise floor below 60 Hz in both detectors, and for H1 yields an appreciable decrease in the 100-450 Hz range.

We demonstrate the gain in sensitivity in continuous wave searches with a Monte Carlo simulation where we consider 500 simulated continuous wave signals with frequency between 20 – 1000 Hz distributed log-uniformly. The amplitude of the signals is such that they are clearly visible in the search results. The signals are added to the real data in the time-domain. The data is then treated as it would be treated for a search, i.e. it is gated and Fourier-transformed in chunks to yield the SFTs. We perform a perfectly matched single-template  $\mathcal{F}$ -statistic search [22] using these SFTs, from non-gated and gated data. We compare the results in Fig. 5. An overall positive effect of gating can be seen, with a relative increase in detection statistic of up to 33% for H1 O1 data. Since gating lowers the noise level more in the low-frequency region, the signal-recovery improves more for low-frequency signals than for higher-frequency signals. The `pyCBC-gating` results are comparable to the `gatestrain` results, so in Figure 5 we only show the `gatestrain` results.

## B. Gating and line-cleaning in the presence of glitches

Gating also mitigates artefacts introduced by glitches at the frequencies cleaned-out in the frequency domain. The line-cleaning procedure used in many continuous

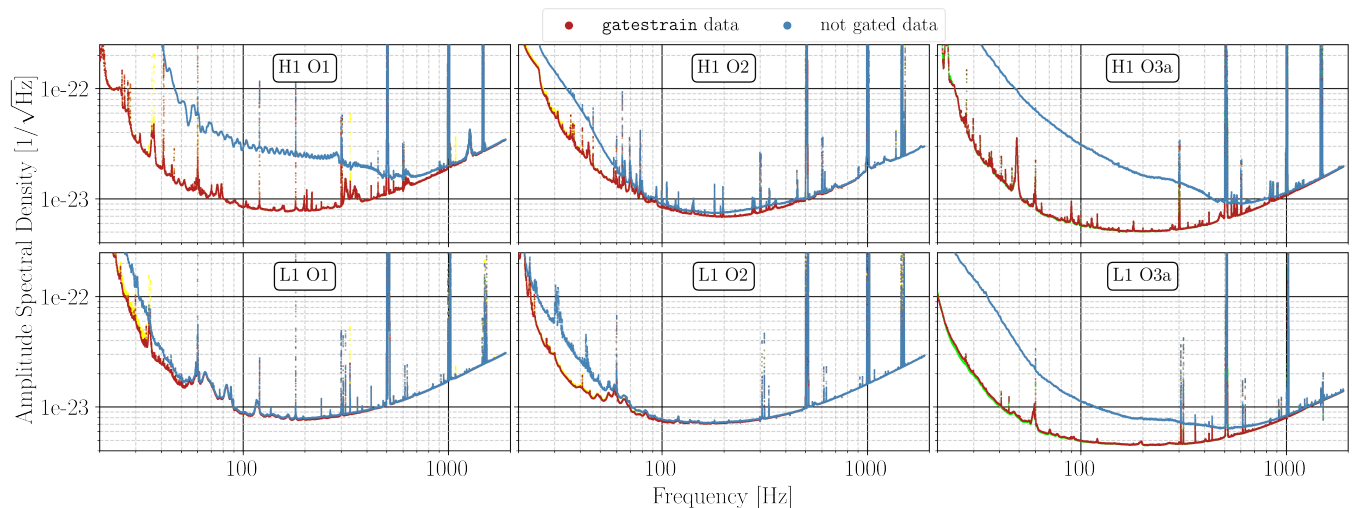


FIG. 4. Average Amplitude Spectral Densities (ASDs) of H1/L1 data from the O1, O2 and O3a runs, before and after removal of glitches with our `gatestrain`-method. The ASD is estimated as the square root of the arithmetic mean over all timestamps, bin per bin, of the power spectral density. For O1 and O2 we also plot the pyCBC-gated data in yellow, and for O3 the ASD of the LVC-gated data [31] in lime green. `gatestrain` achieves a comparable, if not lower, noise floor than these other methods.

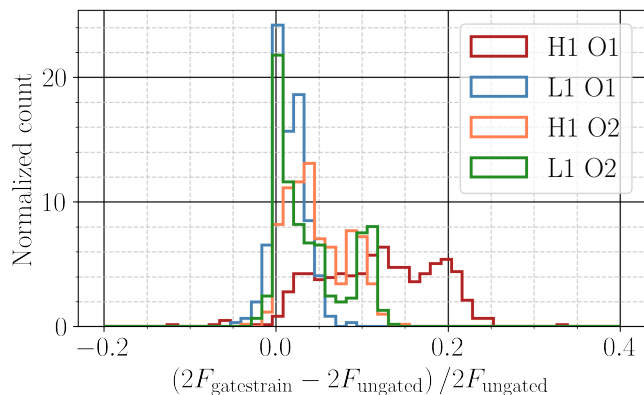


FIG. 5. Relative differences in the detection statistic  $2F$  of 500 recovered simulated signals in `gatestrain`-gated versus non-gated (left panel) SFTs. It can be seen that gating has an overall positive effect which varies depending on detector and observation run.

wave searches substitutes the data at frequency bins that have been flagged to harbour disturbances, with Gaussian noise. In these bins fake SFT data is created with a standard deviation consistent with the noise level estimated based on the real data, in nearby-frequency bins. If the data in these nearby bins is quite Gaussian, the fake noise will look like a realisation of noise from the nearby bins. But if the nearby noise has significant non-gaussian contributions, the fake noise will not look at all like the noise in the nearby bins, and in the presence of loud glitches, it will be higher. The gating removes these non-gaussian contributions and, with them, this type of problem. This is illustrated in Figure 6 that shows how the noise floor of the cleaned data is greatly

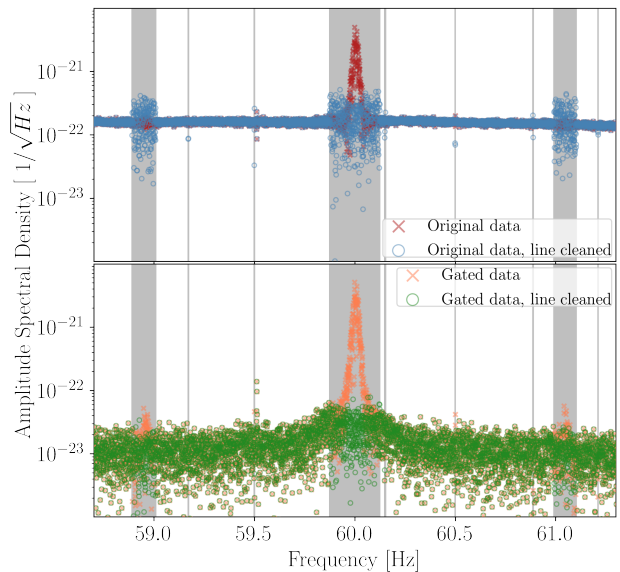


FIG. 6. Amplitude spectral density (ASD) of an SFT disturbed by a loud glitch before and after gating. The top panel shows the ASD of the *original data* before and after line cleaning. The lower panel shows the ASD of the *gated data* with and without line cleaning. The noise floor is greatly reduced thanks to the gating and then line cleaning removes the peaks without the discontinuities evident in the upper plot.

reduced and that the gating before the line-cleaning allows for the lines to be removed without producing other spectral artefacts.

### C. Gating the O3 LIGO data

The first six months of the O3 data (O3a) have just been publicly released. This data presents multiple families of glitches that have required substantial effort by the LVC in order to work-around with an ad-hoc gating procedure [32]. The basic algorithm is simple and it is described in [31]. The resulting gates are released with [31].

We apply our `gatestrain` to the O3a public data with minimal changes in parameters with respect to the O1/O2 data, in order to allow for a slightly more aggressive gating. This is justified because the O3 data is significantly more glitchy than the data from the two previous Advanced LIGO runs. At every iteration we change  $H^{\text{high}}$  by 11, rather than 10; we set the smallest  $H^{\text{high}}$  threshold to be  $H^{\text{low}}$ ; we reduce the threshold for the gating result-check  $R^{\text{th}}$ :  $1.05 \rightarrow 1.01$ .

Table II shows how many SFTs are affected by gates, and how many gates the LVC method and our method produce. It also shows how much data is zeroed-out as a result of these gates. There is a caveat: [31] exclude SFTs with a total gate duration longer than 30s. The data zeroed-out in this way is included in the count given in the last two columns of Table II. In order to make a fair comparison we also adopt this criterium and zero-out the entire SFT when our gate duration is longer than 30s. We include this data in the zeroed-out count in the last columns of Table II for the `gatestrain` method. With this convention `gatestrain` preserves 222 h of data that the LVC gates instead remove.

`gatestrain` achieves a more precise removal: Only 14 (402) SFTs in H1 (L1) are excluded due to long gate duration when using `gatestrain`, while [31] exclude 61 (773) in H1 (L1), respectively. On “good data”, i.e. on SFTs that are free of long-duration gates, the gain is less dramatic, but it is still significant: `gatestrain` removes 3.4 h (10.1 h) in H1 (L1), whereas [31] exclude 8.2 h (13.5 h).

The amplitude spectral densities of the data gated with `gatestrain` and with the LVC method [31], are comparable, as shown in Figure IV A. In both detectors the differences above 40 Hz (150 Hz) are below 1% (0.5%) apart from ranges affected by lines. In these ranges continuous wave searches are difficult to carry out and the data require special treatment, so the performance of this method is not relevant. The largest differences in the achieved noise level are between 20 – 29 Hz in L1 where the `gatestrain` ASD is up to 5% larger than the LVC’s; however in the same frequency range the situation is reversed for H1, where the `gatestrain` ASD is up to 5% smaller than the LVC’s; between 10 – 20 Hz both for H1 and L1 the `gatestrain` ASD is on average 4-5% lower than the LVC’s, with peaks of even 15-20%.

We also recover the fifteen isolated continuous wave hardware injections below 2 kHz [33] using the  $\mathcal{F}$ -statistic with comparable efficiency in both data sets. Like [31] we too could not recover the signal at 12.34 Hz due to the

high noise in this frequency range. The relative gain in detection statistic with respect to [31] is a few percent.

We publish our O3a gates in the Supplemental Materials and at [18].

### V. DISCUSSION

In this paper we present a new method to remove non-Gaussian noise transients in an overall fairly well-behaved noise background. The method extends the gating implementation by [13] with two main novelties: i) the duration of glitches is measured and only the data affected by the glitch is removed ii) the amplitude threshold that defines glitch-data is not fixed, but rather it is adaptive and is iteratively changed during the gating procedure.

As shown in Fig. 1, there is no single typical glitch duration. Glitches come in various sizes and durations, and the glitch populations change from one run to the next. While the `pyCBC` method of [13] is proven perfectly adequate for compact binary coalescence searches in O1 and O2, for continuous wave searches our `gatestrain` method keeps more data untouched. For both H1 and L1 the number of gated SFTs increases slightly compared to the 16 s fixed-gate-duration of `pyCBC` (as used in [28]). On the other hand, `gatestrain` removes less than 10% of the data that `pyCBC` removes.

We note that the `pyCBC` gate duration  $\tau_{\text{dur}} = 16$  s that we used in [28] is cautiously long and thus unsurprisingly more data is lost. In Appendix A we show that using  $\tau_{\text{dur}} = 3$  s does decrease the amount of lost data, but increases the noise in the lower frequencies by up to 7% while still losing 2 ~ 3 times more data compared to `gatestrain`.

Below ~ 1 kHz the recovery of hardware and software injections shows SNR improvements compared to not gating and consistent with what is observed with `pyCBC` gating. No negative effect are recorded apart from regions of loud lines, e.g. violin modes, calibrations lines and power mains. These are cleaned anyway.

Unlike `pyCBC` which is written in Python, `gatestrain` is written in C. It is developed as part of LALSuite and leverages LALSuite functions and methods; it can be found in the LALSuite fork [18] under the name of `lalapps_gateStrain_v1`. It takes ~ 30 – 40 s for an instance of `gatestrain` to produce a gated SFT in the frequency range 10 to 2000 Hz, lasting 1800s. The input data are gravitational wave frame (gwf) files of the public data release. The outputs are gated SFTs or gated gravitational wave frame (gwf) files and optionally ungated SFTs .

Although other gating methods exist [15], they are utilized within specific frameworks, the software is not publicly available and the input data is not the standard gravitational wave frame (gwf) format. Our `lalapps_gateStrain_v1` works within the general LIGO Algorithm Library framework, and could be in fact be merged in the official LALSuite repository.

We present the results in the context of continuous gravitational wave searches, however the method is valid for other searches, e.g. transient searches and stochastic background searches. We analyze the data around the eleven compact binary coalescence gravitational wave events of the GWTC-1 catalog. Not one was gated with out-of-the-box `gatestrain`-method. The glitch near GW170817 was automatically detected starting 1.09 s before the event, 7.41 ms off of LIGO’s gate mid-time. Our `gatestrain` applied a 92.65 ms gate with 0.25 s taper to each side in comparison to LIGO’s gate with 0.2 s duration and 0.5 s taper [17, 30]. As the rate of short-duration signal detections increases, an efficient method that automatically excises the disturbed portions of the data, becomes important.

Borne out of the desire to generalize the methodology that we had used on O1 data, and successfully applied to O2 data, our gating procedure successfully gates the O3 data, achieving a much smaller data loss than the LIGO gates with the same spectral noise improvement. Our procedure removes less than half the data compared to the ad-hoc gating of [31].

Since the gated data remains a small fraction of the total data set, the impact of the more efficient gating on the detection statistic is small and the very loud hardware-injected fake signals present in the LIGO data for validation purposes, are recovered with comparable values of the detection statistic in both `gatestrain` data and LVC-gated data. The benefits of our method for gating is that it does not require ad-hoc time-consuming studies and careful tuning for every new data set and every new family of glitches that appears.

Thanks to its adaptive algorithm, with practically no tuning, we were able to determine the O3a gates in less than a week. We make our tool available together with the O3a gates in the supplemental material [18], for others to employ in their analyses of LIGO O3 data. [18] will be updated with the O3b gates, as soon as that data becomes public.

## ACKNOWLEDGMENTS

We thank Alex Nitz, Tito Dal Canton and Badri Krishnan for useful discussions on the `pyCBC` gating. This work has utilised the ATLAS cluster computing at MPI for Gravitational Physics Hannover. This research

has made use of data, software and/or web tools obtained from the Gravitational Wave Open Science Center (<https://www.gw-openscience.org>), a service of LIGO Laboratory, the LIGO Scientific Collaboration and the Virgo Collaboration. LIGO is funded by the U.S. National Science Foundation. Virgo is funded by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale della Fisica Nucleare (INFN) and the Dutch Nikhef, with contributions by Polish and Hungarian institutes.

## Appendix A: `pyCBC` gating with a 3 s gate duration

In this paper we compare the performance performance of our `gatestrain` with that of the `pyCBC` method with a fixed gate duration of 16 s. The reason is that, in absence of non-fixed duration gating procedures, 16 s is the `pyCBC` gate duration that was used in previous continuous wave searches [28]. The typical `pyCBC` gate duration for compact binary coalescence searches is 3 s. So, while our 16 s choice removed more data than a transient signal search would remove, it still removed a very small portion of the data and did not impact the continuous wave search sensitivity. It may however be interesting to compare it to `pyCBC`-gating with 3 s gates. This is shown in Tab. III for the O1 and O2 data. Not surprisingly less data is lost with the 3 s gate duration compared to 16 s, but still  $\sim 3$  times more data is lost in comparison with our `gatestrain` method. Fig. 7 shows the relative difference of average PSDs created from data gated with 16 s and with 3 s `pyCBC` fixed-duration gating. While these perform similarly overall, in the low frequency region the 16 s `pyCBC` gating outperforms the `pyCBC` 3 s gating by of order 10%.

DURATION OF GATED DATA			
	<code>pyCBC</code> 16 s	<code>pyCBC</code> 3 s	<code>gatestrain</code>
	[h]	[h]	[h]
H1	3.43	0.68	0.23
L1	0.86	0.17	0.08

TABLE III. Amount of gated O1 data with different `pyCBC` gate-duration values. 3 s is the parameter value used in compact binary coalescence searches; 16 s is the value that was used in previous continuous wave searches [28].

[1] D. Davis *et al.* [LIGO], “LIGO Detector Characterization in the Second and Third Observing Runs,” [arXiv:2101.11673 [astro-ph.IM]].

[2] F. Robinet, N. Arnaud, N. Leroy, A. Lundgren, D. Macleod and J. McIver, “Omicron: a tool to characterize transient noise in gravitational-wave detectors,” [arXiv:2007.11374 [astro-ph.IM]].

[3] B. P. Abbott *et al.* [LIGO Scientific and Virgo], Class. Quant. Grav. **37** (2020) no.5, 055002 doi:10.1088/1361-6382/ab685e [arXiv:1908.11170 [gr-qc]].

[4] C. Pankow, K. Chatziioannou, E. A. Chase, T. B. Littenberg, M. Evans, J. McIver, N. J. Cornish, C. J. Haster, J. Kanner and V. Raymond, *et al.* “Mitigation of the instrumental noise transient in gravitational-wave data surrounding GW170817,” Phys. Rev. D **98**



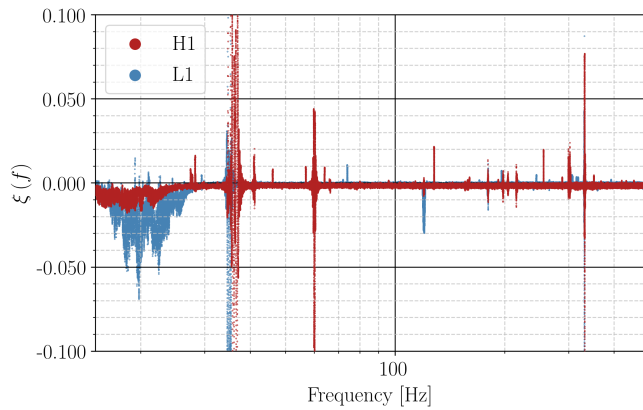


FIG. 7.  $\xi(f) = \frac{(\mathcal{P}_{16s} - \mathcal{P}_{3s})}{\mathcal{P}_{16s}}$ , relative difference in the noise PSD  $\mathcal{P}$  when using data gated with a 3 s fixed-duration gate and with a 16 s gate (pyCBC). The 16 s gate duration produces overall a slightly lower noise floor and up to 7% lower in the lowest frequency range. We note that the outliers are associated with spectral lines and these are cleaned-out with a separate procedure. The gating procedure aims at lowering the noise floor.

- (2018) no.8, 084016 doi:10.1103/PhysRevD.98.084016 [arXiv:1808.03619 [gr-qc]].
- [5] J. C. Driggers *et al.* [LIGO Scientific], “Improving astrophysical parameter estimation via offline noise subtraction for Advanced LIGO,” *Phys. Rev. D* **99** (2019) no.4, 042001 doi:10.1103/PhysRevD.99.042001 [arXiv:1806.00532 [astro-ph.IM]].
- [6] B. P. Abbott *et al.* [LIGO Scientific and Virgo], “Characterization of transient noise in Advanced LIGO relevant to gravitational wave signal GW150914,” *Class. Quant. Grav.* **33** (2016) no.13, 134001 doi:10.1088/0264-9381/33/13/134001 [arXiv:1602.03844 [gr-qc]].
- [7] J. L. McIver, “The impact of terrestrial noise on the detectability and reconstruction of gravitational wave signals from core-collapse supernovae,” *UMASS-539*.
- [8] B. P. Abbott *et al.* [LIGO Scientific and Virgo], “GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs,” *Phys. Rev. X* **9** (2019) no.3, 031040 doi:10.1103/PhysRevX.9.031040
- [9] R. Abbott *et al.* [LIGO Scientific and Virgo], “GWTC-2: Compact Binary Coalescences Observed by LIGO and Virgo During the First Half of the Third Observing Run,” [arXiv:2010.14527 [gr-qc]].
- [10] A. H. Nitz, C. Capano, A. B. Nielsen, S. Reyes, R. White, D. A. Brown and B. Krishnan, “1-OGC: The first open gravitational-wave catalog of binary mergers from analysis of public Advanced LIGO data,” *Astrophys. J.* **872** (2019) no.2, 195 doi:10.3847/1538-4357/ab0108
- [11] A. H. Nitz, T. Dent, G. S. Davies, S. Kumar, C. D. Capano, I. Harry, S. Mozzon, L. Nuttall, A. Lundgren and M. Tápai, “2-OGC: Open Gravitational-wave Catalog of binary mergers from analysis of public Advanced LIGO and Virgo data,” *Astrophys. J.* **891**, 123 doi:10.3847/1538-4357/ab733f
- [12] A. H. Nitz, C. D. Capano, S. Kumar, Y. F. Wang, S. Kasta, M. Schäfer, R. Dhurkunde and M. Cabero, “3-OGC: Catalog of gravitational waves from compact-binary mergers,” [arXiv:2105.09151 [astro-ph.HE]].
- [13] S. A. Usman, A. H. Nitz, I. W. Harry, C. M. Biwer, D. A. Brown, M. Cabero, C. D. Capano, T. Dal Canton, T. Dent and S. Fairhurst, *et al.* “The PyCBC search for gravitational waves from compact binary coalescence,” *Class. Quant. Grav.* **33**, no.21, 215004 (2016) doi:10.1088/0264-9381/33/21/215004
- [14] P. Astone, S. Frasca and C. Palomba, “The short FFT database and the peak map for the hierarchical search of periodic sources,” *Class. Quant. Grav.* **22**, S1197-S1210 (2005) doi:10.1088/0264-9381/22/18/S34
- [15] P. Leaci, P. Astone, M. A. Papa and S. Frasca, “Using a cleaning technique for the search of continuous gravitational waves in LIGO data,” *J. Phys. Conf. Ser.* **228**, 012006 (2010) doi:10.1088/1742-6596/228/1/012006
- [16] P. B. Covas *et al.* [LSC], “Identification and mitigation of narrow spectral artifacts that degrade searches for persistent gravitational waves in the first two observing runs of Advanced LIGO,” *Phys. Rev. D* **97**, no.8, 082002 (2018) doi:10.1103/PhysRevD.97.082002
- [17] M. Vallisneri, J. Kanner, R. Williams, A. Weinstein and B. Stephens, *J. Phys. Conf. Ser.* **610** (2015) no.1, 012021 doi:10.1088/1742-6596/610/1/012021
- [18] [www.aei.mpg.de/continuouswaves/Gating2021](https://www.aei.mpg.de/continuouswaves/Gating2021)
- [19] <https://doi.org/10.7935/CA75-FM95> LIGO Open Science Center, <https://losc.ligo.org>.
- [20] [doi.org/10.7935/CA75-FM95](https://doi.org/10.7935/CA75-FM95) LIGO Open Science Center, <https://losc.ligo.org>.
- [21] B. Allen and G. Mendell, “SFT Data Format Version 2 Specification”, (2004), <https://dcc.ligo.org/LIGO-T040164/public>
- [22] P. Jaranowski, A. Krolak and B. F. Schutz, “Data analysis of gravitational - wave signals from spinning neutron stars. 1. The Signal and its detection,” *Phys. Rev. D* **58** (1998) 063001 doi:10.1103/PhysRevD.58.063001 [gr-qc/9804014].
- [23] O. J. Piccinni, P. Astone, S. D’Antonio, S. Frasca, G. Intini, P. Leaci, S. Mastrogiovanni, A. Miller, C. Palomba and A. Singhal, “A new data analysis framework for the search of continuous gravitational wave signals,” *Class. Quant. Grav.* **36** (2019) no.1, 015008 doi:10.1088/1361-6382/aaefb5 [arXiv:1811.04730 [gr-qc]].
- [24] R. Abbott *et al.* [LIGO Scientific, Virgo and KAGRA], “Upper Limits on the Isotropic Gravitational-Wave Background from Advanced LIGO’s and Advanced Virgo’s Third Observing Run,” [arXiv:2101.12130 [gr-qc]].
- [25] Y. Zhang, M. A. Papa, B. Krishnan and A. L. Watts, “Search for Continuous Gravitational Waves from Scorpius X-1 in LIGO O2 Data,” *Astrophys. J. Lett.* **906** (2021) no.2, L14 doi:10.3847/2041-8213/abd256 [arXiv:2011.04414 [astro-ph.HE]].
- [26] R. Abbott *et al.* [LIGO Scientific and Virgo], “All-sky search in early O3 LIGO data for continuous gravitational-wave signals from unknown neutron stars in binary systems,” *Phys. Rev. D* **103** (2021) no.6, 064017 doi:10.1103/PhysRevD.103.064017 [arXiv:2012.12128 [gr-qc]].
- [27] M. A. Papa, J. Ming, E. V. Gotthelf, B. Allen, R. Prix, V. Dergachev, H. B. Eggenstein, A. Singh and S. J. Zhu, “Search for Continuous Gravitational Waves from the Central Compact Objects in Supernova Remnants Cassiopeia A, Vela Jr., and G347.3-0.5,” *Astrophys. J.* **897** (2020) no.1, 22 doi:10.3847/1538-

- 4357/ab92a6 [arXiv:2005.06544 [astro-ph.HE]].
- [28] J. Ming, M. A. Papa, A. Singh, H. B. Eggenstein, S. J. Zhu, V. Dergachev, Y. M. Hu, R. Prix, B. Machenschalk and C. Beer, *et al.* “Results from an Einstein@Home search for continuous gravitational waves from Cassiopeia A, Vela Jr. and G347.3,” *Phys. Rev. D* **100** (2019) no.2, 024063 doi:10.1103/PhysRevD.100.024063 [arXiv:1903.09119 [gr-qc]].
- [29] B. Steltner, M. A. Papa, H. B. Eggenstein, B. Allen, V. Dergachev, R. Prix, B. Machenschalk, S. Walsh, S. J. Zhu and S. Kwang, “Einstein@Home All-sky Search for Continuous Gravitational Waves in LIGO O2 Public Data,” *Astrophys. J.* **909** (2021) no.1, 79 doi:10.3847/1538-4357/abc7c9 [arXiv:2009.12260 [astro-ph.HE]].
- [30] B. P. Abbott *et al.* [LIGO Scientific and Virgo], *Phys. Rev. Lett.* **119** (2017) no.16, 161101 doi:10.1103/PhysRevLett.119.161101 [arXiv:1710.05832 [gr-qc]].
- [31] J. Zweizig and K. Riles “Information on self-gating off(t)used in O3 continuous-wave searches”, (2021) Technical Document T2000384 available at <https://dcc.ligo.org/>
- [32] J. Wang and K. Riles, LIGO Scientific Collaboration, “A Semi-Coherent Directed Search for Continuous Gravitational Waves from Supernova Remnants in the LIGO O3 Data Set”, presentation at the April 2021 APS meeting, session Y16.00007.
- [33] [https://www.gw-openscience.org/O3/O3April1\\_injection\\_parameters](https://www.gw-openscience.org/O3/O3April1_injection_parameters)