Application of dictionary learning to denoise LIGO's blip noise transients

Alejandro Torres-Forné,[1,*] Elena Cuoco,[2,3,4] José A. Font,[5,6] and Antonio Marquina[7]

[1]*Max Planck Institute for Gravitational Physics (Albert Einstein Institute),*
D-14476 Potsdam-Golm, Germany
[2]*European Gravitational Observatory (EGO), Via E Amaldi, I-56021 Cascina, Italy*
[3]*Scuola Normale Superiore (SNS), Piazza dei Cavalieri, 7, 56126 Pisa PI, Italy*
[4]*Istituto Nazionale di Fisica Nucleare (INFN) Sez.*
Pisa Edificio C—Largo B. Pontecorvo 3, 56127 Pisa, Italy
[5]*Departamento de Astronomía y Astrofísica, Universitat de València,*
Dr. Moliner 50, 46100 Burjassot (València), Spain
[6]*Observatori Astronòmic, Universitat de València,*
C/ Catedrático José Beltrán 2, 46980 Paterna (València), Spain
[7]*Departamento de Matemática Aplicada, Universitat de València,*
Dr. Moliner 50, 46100 Burjassot (València), Spain

(Received 26 February 2020; accepted 24 June 2020; published 8 July 2020)

Data streams of gravitational-wave detectors are polluted by transient noise features, or "glitches," of instrumental and environmental origin. In this work we investigate the use of total variation methods and learned dictionaries to mitigate the effect of those transients in the data. We focus on a specific type of transient, "blip" glitches, as this is the most common type of glitch present in the LIGO detectors and their waveforms are easy to identify. We randomly select 100 blip glitches scattered in the data from advanced LIGO's O1 run, as provided by the citizen-science project Gravity Spy. Our results show that dictionary-learning methods are a valid approach to model and subtract most of the glitch contribution in all cases analyzed, particularly at frequencies below ~1 kHz. The high-frequency component of the glitch is best removed when a combination of dictionaries with different atom length is employed. As a further example we apply our approach to the glitch visible in the LIGO-Livingston data around the time of merger of binary neutron star signal GW170817, finding satisfactory results. This paper is the first step in our ongoing program to automatically classify and subtract all families of gravitational-wave glitches employing variational methods.

DOI: 10.1103/PhysRevD.102.023011

I. INTRODUCTION

The third observational campaign of the advanced gravitational-wave (GW) detectors LIGO [1] and Virgo [2], O3, is currently ongoing. During the previous two campaigns, O1 and O2, the GW detector network reported the observation of eleven compact binary mergers [3] comprising ten binary black holes and one binary neutron star. The latter, GW170817 [4], was accompanied by extensive and very successful follow-up observations of electromagnetic emission originated from the same astronomical source [5]. Moreover, neutrino searches were also carried out, yet no detection has been reported [6]. The entire GW strain data from O1/O2 has been made publicly available and, in particular, the data around the time of each of the eleven O1/O2 events are accessible through the Gravitational-Wave Open Science Center.[1] Since the start of O3 on April 1st 2019, GW candidate events are being released as public alerts to facilitate the rapid identification of electromagnetic or neutrino counterparts. The growing list of candidates can be inspected at the GW Candidate Event Database [7]. Toward the end of O3, the GW detector network may be increased by yet another facility with the addition of the KAGRA detector [8].

The detection of GWs is severely hampered by many sources of noise that contribute to a nonstationary background in the time series of data in which actual GW signals reside. The *theoretical* sensitivity of the instruments is limited at low frequencies (below ~20 Hz) by gravity-gradient (seismic) noise and at high frequencies

[*] Corresponding author.
alejandro.torres-forne@aei.mpg.de

[1] `https://gw-openscience.org`

(above ∼2 kHz) by photon-shot noise originated by quantum fluctuations of the laser. The detectors are most sensitive at intermediate frequencies (∼200 Hz) where the Brownian motion of the suspensions and mirrors is the limiting source of so-called thermal noise [9]. *While these theoretical limits have not yet been entirely satisfied, they can be anticipated to be limiting in future observing runs. At present, the most limiting source of noise at low frequencies is control noise.* Moreover, the data stream is polluted with the presence of transient (short duration) noise signals, commonly known as "glitches," whose origin is not astrophysical but rather instrumental and environmental. We refer to [10] for a comprehensive overview of the LIGO/Virgo detector noise and the extraction of GW signals.

Glitches *impede* GW data analysis for a number of reasons. By their short-duration nature they contribute significantly to the background of transient GW searches. Glitches may occur sufficiently frequently to potentially affect true signals, particularly when occurring in (or almost in) coincidence. Furthermore, some types of glitches show time-frequency morphologies remarkably similar to actual transient astrophysical signals, which increases the false-alarm rate of potential triggers. Moreover, having to remove portions of data in which glitches are present downgrades the duty cycle of the detectors. It is however not trivial to remove defective segments of data. The simplest approach, i.e., setting them to zero, might result in a leakage of excess power, which may turn the mitigating approach more damaging than the very effect of the glitch. Traditional automated approaches to discard triggers, like that of [11], analyze auxiliary channels from the detectors which are insensitive to GW signals. These approaches efficiently produce vetoes ranking the statistical relationship between triggers in auxiliary channels and remove the corresponding data in the GW channel.

For all these reasons, understanding the origin of glitches and mitigating their effects is a major effort in the characterization of GW detectors [12,13]. Indeed, in recent years many strategies have been developed to automatically classify glitches. The approaches are as diverse as Bayesian inference, machine learning, deep learning, and citizen science [14–22]. Recent examples of glitch mitigation are reported in [23–26]. Reference [23] describes various deglitching methods to extract the strong glitch present in the LIGO-Livingston detector about 1s before the merger of the binary neutron star that produced the signal GW170817 [4]. In [24,25] the impact of loud glitches is reduced by using an inpainting filter that fills the hole created after windowing the glitch. Glitch reduction, together with other techniques, was shown to improve the statistical significance of a GW trigger. In addition, deep learning approaches have also proven very effective to recover the true GW signal even in the presence of glitches [26].

There are many different families of glitches identified during the advanced LIGO-Virgo observing runs [16,27,28]. Glitches from each family have a similar morphology, although the characteristics of each specific glitch in terms of duration, bandwidth and signal-to-noise (SNR) ratio can vary significantly even for glitches inside the same family. In this work we focus on blip glitches, a noise transient characterized by a duration of about 10 ms and a frequency bandwidth of about 100 Hz. This type of *glitch*, which has mainly been found in the two LIGO detectors, significantly reduce the sensitivity of searches for high-mass compact binary coalescences. Blip glitches in LIGO data are identified using both the PyCBC pipeline search (see [27] and references therein) and the citizen-science effort Gravity Spy [16]. PyCBC is a matched filter technique that uses short-duration compact binary coalescence templates while Gravity Spy uses the time-frequency morphology to classify glitches among different families. The recent study of [27] based on PyCBC has found that Advanced LIGO data during O1/O2 contains approximately two blip glitches per hour of data (amounting to thousands of blip glitches in total). The physical origin of most of them remains unclear. This paper explores the performance of dictionary-learning methods [29] to mitigate the presence of blip glitches in advanced LIGO data. To this aim we select a large number of blip glitches randomly distributed along the data stream from advanced LIGO's first observing run. For each glitch, the data correspond to a one-second window centered at the GPS time of the glitch as provided by Gravity Spy [16]. As in [23,24,26] the goal of our work is to mitigate the impact of glitches in GW data in order to increase the statistical significance of astrophysical triggers. Our results show that dictionary-learning techniques are able to model blip glitches and to subtract them from the data without significantly disturbing the background.

The paper is organized as follows: In Sec. II we summarize the mathematical framework of the variational methods which are at the core of the dictionary-learning approach we use. Section III discusses technical aspects, namely the whitening procedure we employ to remove noise lines and other artefacts, the training of the dictionaries, and how we perform the reconstruction of the glitches. The results of our study are presented in Sec. IV. Finally, a summary is provided in Sec. V. Appendix shows the spectrograms of the 16 blip glitches from O1 we employ in our test set and reports their main characteristics.

II. REVIEW OF L_1-NORM VARIATIONAL METHODS

A. Total variation methods

In this paper we employ two different variational techniques based on the L_1 norm, the Rudin-Osher-Fatemi (ROF) method [30] and a dictionary learning method [29]. We have recently begun to use these procedures in the context of GW data analysis in [20,31–33]. Both approaches solve the denoising problem, $y = u + n$, where u is the true signal and n is the noise, as a variational problem. The solution u is thus obtained as

$$u_\lambda = \underset{u}{\operatorname{argmin}}\left\{\mathcal{R}(u) + \frac{\lambda}{2}\mathcal{F}(u)\right\}, \tag{2.1}$$

where $\mathcal{R}$ is the regularization term, i.e., the constraint to impose in the data and $\mathcal{F}$ is the fidelity term, which measures the similarity of the solution to the data. The parameter λ is the regularization parameter and controls the relative weight of both terms in the equation. Even though both methods solve the same general problem, each one of them approaches the problem in a different way and, therefore, the regularization term and the fidelity term have different expressions.

In 1992, Rudin, Osher and Fatemi (ROF) [30] proposed the use of the so-called total-variation (TV) norm as the regularization term $\mathcal{R}(u) = \int_\Omega |\nabla u|$ constrained to $\|y - u\|^2$. Note that $|\cdot|$ and $\|\cdot\|$ represent the L_1 and L_2 norms, respectively. This specific formulation of the variational problem (2.1) is called the ROF model and reads

$$u_\lambda = \underset{u}{\operatorname{argmin}}\left\{\int_\Omega |\nabla u| + \frac{\lambda}{2}\|y - u\|^2\right\}. \tag{2.2}$$

This model preserves steep gradients, reduces noise by *sparsifying* (i.e., promoting zeros) the gradient of the signal and avoids spurious oscillations (Gibbs effect). However, the associated Euler-Lagrange equation, given by

$$\nabla\cdot\frac{\nabla u}{|\nabla u|} + \lambda(y - u) = 0, \tag{2.3}$$

becomes singular when $|\nabla u| = 0$. This issue can be easily solved by changing the standard TV norm by a slightly perturbed version (see [31] for a detailed explanation),

$$\mathrm{TV}_\beta(u) \coloneqq \int \sqrt{|\nabla u| + \beta}, \tag{2.4}$$

where β is a small positive parameter. We refer to this modified version of the regularization term in the method as regularized ROF (rROF).

B. Sparse reconstruction over a fixed dictionary

In dictionary-based methods, the denoising is performed by assuming that the true signal u can be represented as a linear combination of the columns (atoms) of a matrix $\boldsymbol{D}$ called the dictionary. If the signal can be represented with a *few* columns of $\boldsymbol{D}$, the dictionary is adapted to u. In other words, there exists a "sparse vector" α such that $u \sim \boldsymbol{D}\alpha$. As a result, the fidelity term in Eq. (2.1) reads,

$$\mathcal{F}(\alpha) = \|y - \boldsymbol{D}\alpha\|^2. \tag{2.5}$$

In other words, the problem reduces to finding a sparse vector α that represents the signal u over the columns of the dictionary. The next step is to find a regularization term that induces sparsity over the coefficients of α. Classical dictionary-learning techniques [34,35] use as regularization term the L_0-norm, which is chosen to ensure that the solution has the fewest possible number of nonzero coefficients. However, this problem is not convex and is NP-hard, i.e., it can be solved in nondeterministic polynomial time. If the L_1-norm is used instead of the L_0-norm, the problem becomes convex. This type of regularization promotes zeros in the components of the vector coefficient α, and the solution is the sparsest one in most cases. The variational problem thus reads,

$$\alpha_\lambda = \underset{\alpha}{\operatorname{argmin}}\left\{|\alpha| + \frac{\lambda}{2}\|\boldsymbol{D}\alpha - y\|^2\right\}, \tag{2.6}$$

which is known as *basis pursuit* [36] or *LASSO* [37]. In this paper we solve Eq. (2.6) using the alternating direction method of multipliers (ADMM) algorithm [38].

C. Dictionary learning

In the previous section we have assumed that the dictionary $\boldsymbol{D}$ is fixed and we only solve the problem of representation. Traditionally, predefined dictionaries based on wavelets, curvelets, etc., have been used. However, signal reconstruction can be dramatically improved by learning the dictionary instead of using a predefined one [39]. In this approach a set of training signals is divided into patches in such a way that the length of the patches is less than the total length of the training signals. In most common problems, the number of training patches m is large compared with the length of each patch n, $n \ll m$. The procedure to train the dictionary is similar to Eq. (2.6) except that the dictionary $\boldsymbol{D}$ should now be added as variable,

$$\alpha_\lambda, \boldsymbol{D}_\lambda = \underset{\alpha,\boldsymbol{D}}{\operatorname{argmin}}\left\{\frac{1}{n}\sum_{i=1}^{m}\|\boldsymbol{D}\alpha_i - x_i\|_2^2 + \lambda|\alpha_i|\right\}, \tag{2.7}$$

where x_i denotes the ith training patch. Unfortunately, this problem is not jointly convex unless the variables are considered separately. In [29] a method based on stochastic approximations was proposed. These approximations process one sample at a time and the method takes advantage of the problem structure to efficiently solve it. For each element in the training set, the algorithm alternates a classical sparse coding step, to solve for α using a dictionary $\boldsymbol{D}$ obtained in the previous iteration, with a dictionary update step, where the new dictionary is calculated with the recently calculated values of α, namely

$$\alpha^{k+1} = \underset{\alpha}{\mathrm{argmin}}\left\{\frac{1}{n}\sum_{i=1}^{m} \|\boldsymbol{D}^k \alpha_i - \mathbf{u}_i\|^2 + \lambda|\alpha_i|\right\} \tag{2.8}$$

$$\boldsymbol{D}^{k+1} = \underset{\boldsymbol{D}}{\mathrm{argmin}}\left\{\frac{1}{n}\sum_{i=1}^{m} \|\boldsymbol{D}\alpha_i^{k+1} - \mathbf{u}_i\|^2 + \lambda|\alpha_i|\right\} \tag{2.9}$$

As in [29] we use a block-coordinate descent method [40] for solving $\boldsymbol{D}$ and α_i iteratively.

III. DATA SELECTION AND DICTIONARY GENERATION

In our previous work we applied learned dictionaries to denoise numerically generated gravitational waveforms from simulations of supernovae core-collapse and binary black hole mergers [32] and to classify simulated glitches [20]. The data employed was in either case embedded in nonwhite Gaussian noise to simulate the background noise of advanced LIGO in its broadband configuration. This work takes a step further in our efforts by tackling the denoising problem with dictionaries employing real data, in the form of actual glitches from advanced LIGO's O1 data.

We focus on blip glitches, the most common type of glitch found in the two LIGO detectors and whose origin remains mostly unknown [27]. Blip glitches, characterized by a duration of ~10 ms and a frequency bandwidth of ~100 Hz, have a distinctive tear-drop shape morphology when seen is a time-frequency (spectrogram) plot. By their intrinsic properties and recurring presence they can significantly reduce the sensitivity of searches for high-mass compact binary coalescences. In order to focus only in these noise transients, we apply a whitening procedure to the O1 data to remove all systematic sources of noise from the data, like the calibration and control signals that appear as lines in the spectrum, and also to flatten the data in frequency. The whitening algorithm uses the autoregressive (AR) model of [41,42]. The AR model employs 3000 coefficients estimated using 300 s of data at the beginning of the corresponding science segment of every glitch. This type of whitening has proved to be very robust and, as it is applied in the time domain, it is not affected by the typical border problems that appear in frequency-domain methods.

To measure the accuracy of the glitch reconstruction and mitigation we employ two quantitative estimators. The first one is based on the time-frequency distribution of the power of the signal. We integrate the power spectrum for all frequencies for each temporal bin, and then we calculate the ratio between the maximum power and the mean power for all times. We will refer to this estimator as SNR:

$$\mathrm{SNR} = \frac{\max(S(t))}{\overline{S(t)}}, \qquad S(t) = \int_{20}^{\frac{f_S}{2}} S(t,f)df, \tag{3.1}$$

where $S(t,f)$ is the time-frequency representation of the data, and f_s is the sampling frequency. This SNR estimator is different from the one provided by the optimal filter, which is based in theoretical templates. Our second estimator is called the "whiteness" [42]

$$W = \frac{\exp(1/f_s \int_{-f_s/2}^{f_s/2} \ln(P(f))df)}{1/f_s \int_{-f_s/2}^{f_s/2} P(f)df}, \tag{3.2}$$

where $P(f)$ is the power spectral density (PSD) of the data. The whiteness measures the spectral flatness. As the presence of glitches implies an increment of power with respect to a glitchfree background, if $P(f)$ is very peaky then $W \sim 0$, and if $P(f)$ is flat then $W = 1$.

A. Training

We use the set of blip glitches identified by Gravity Spy in the data from advanced LIGO's first observing run [16] and randomly select 100 blip glitches. This sample, despite not being too large, is sufficient to assess the performance of learned dictionaries in removing glitches while at the same time is computationally cheap. In contrast to other machine-learning approaches (e.g., deep learning) which have hundreds or thousands of parameters to adjust, dictionary-learning techniques do not require a large number of training examples to produce satisfactory results. The data corresponds to a window of 1 s centered at the GPS time of the glitch as provided by Gravity Spy. The data is divided in two different sets; 85% is used to train the dictionary while the remaining 15% (which includes 16 blip glitches) is used to test the algorithm. The morphologies of all 16 glitches are presented in Appendix. The data is downsampled from their original 16384 Hz to 8192 Hz to speed up the algorithm and reduce the computational cost.

The training process is performed as follows. After whitening all the data, we select the main glitch morphology using a window of 1024 samples around the GPS time of the glitch. Then, data from all glitches are aligned and organized in a matrix to build the initial dictionary. Next, we select 30000 random patches of a given length and start the block-coordinate descend method to obtain the trained dictionary. The length of the patches is the same as the atoms of the dictionary and, jointly with the number of atoms, is a hyperparameter of the model.

We explore dictionaries formed by atoms of length in the range $[2^3, 2^9]$. We also vary the number of atoms of each length to understand its possible effect on the results. Our study shows that a dictionary of 128 samples for atoms is a good choice. The number of atoms seems not to be very relevant as long as the dictionary is overcompleted, i.e., the number of atoms is larger than the length of the atoms.

One intrinsic difficulty of applying dictionary-reconstruction techniques to noise transients instead of to actual signals is that we lack a "clean" signal to use as a model to the dictionary. Glitches have a random component due to the background. Even though the learning step has

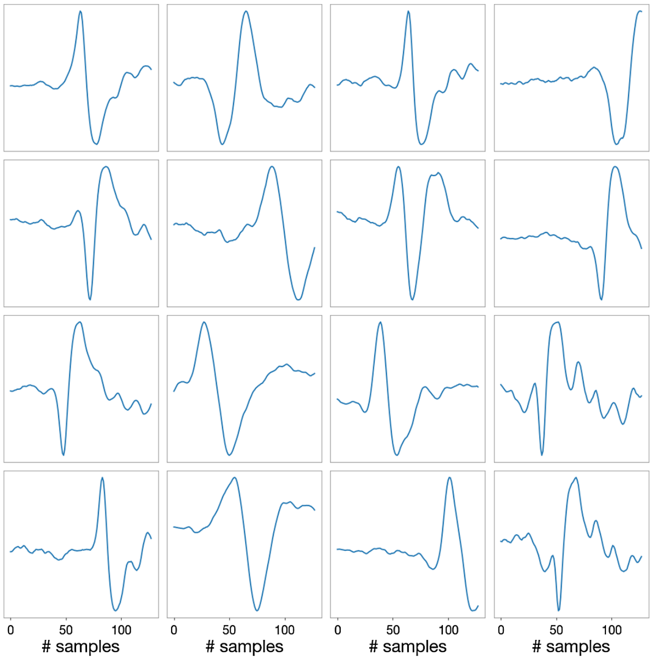

FIG. 1. Example of a dictionary composed by a total of 192 atoms with 128 samples each. Only 16 atoms randomly selected are shown.

denoising capabilities, the block-coordinate descend method can have problems of convergence when the patches contain a large stochastic component. To improve the convergence of the learning step and the extraction results, we introduce an additional step between the whitening and the patch extraction. Namely, we use the rROF method to reduce the variance of the data used for training. This process results in smoother atoms and in a cleaner reconstruction of the glitch, which translates in a better separation between the background and the glitch morphology. An example of a dictionary is shown in Fig. 1.

B. Reconstruction

Once the training step is complete, we use the resulting dictionary to extract the blip glitch from the background. As the length of the atoms is always shorter than the length of the test signals, we perform the reconstruction with a sliding window with an overlap of $n-4$ samples, where n is the length of the atoms. The overlapped samples are averaged to obtain the final reconstruction.

In addition, we apply the ADMM algorithm in an iterative way. Starting with the original data y, we perform a reconstruction over the dictionary u. Then, this reconstructed signal is subtracted from the original data and the resulting residual is used as the new input. This procedure converges in the sense that in each iteration we subtract less signal from the background. Therefore, this iterative process is applied until the differences between the residuals of consecutive iterations is less than a given tolerance. For most cases, a typical tolerance of 10^{-3}–10^{-4} is enough to produce good results.

C. Regularization parameter search

Reconstruction results heavily depend on the value of the regularization parameter λ. If its value is large, the relative weight of the regularization term in Eq. (2.6) is larger and more atoms are used. On the contrary, with a low value of λ less atoms are used and more details of the signal (and noise) are recovered. In previous papers [31–33], we found the optimal value, i.e., the one that produces the best results, comparing the denoised signal with the original one from GW catalogs from numerical relativity. In the present case, as there is not a true signal to compare with, we cannot determine the optimal λ in the same way.

However, as our goal is to extract the glitch trying to keep the background unaltered, we design a different approach to obtain an optimal value of λ. One of the good properties of dictionary-learning methods we found in our previous studies is that the reconstruction returns zero when the data are very different from the dictionary. This statement is true for sufficiently large values of λ. As mentioned before, if the value is too low, the regularization term in Eq. (2.6) becomes negligible and we would be solving essentially a least mean-squares fit, which in practice translates in a very oscillating reconstruction. If the value is very large, it is the fidelity term the one that becomes negligible, and the problem transforms in the minimization of the L_1-norm of the vector α, whose solution is the vector zero. Therefore, our goal is to find the first value of λ that returns zeros in the part of the data dominated by the background but also produces a reconstruction for the glitch. The resulting reconstruction will be zero on the window border and will avoid discontinuities. We select a small window at the beginning of the data window where we are sure that the data is mostly dominated by the background. After that, a bisection algorithm tries to find the largest value that returns a nonzero reconstruction. We will refer to this value as $\lambda_{\min}$.

IV. RESULTS

A. Blip glitch subtraction with a single dictionary

We start applying dictionaries of different length and number of atoms to the 16 blip glitches we use as tests. The bisection procedure introduced in Sec. III C is used to find the lower value of λ that returns zeros at the beginning of the data stream. The reconstructed glitch is then subtracted from the data to obtain a cleaner background.

As mentioned previously, we explore the results of using dictionaries formed by atoms of length inside the range $[2^3, 2^9]$. All options are able to reconstruct the glitches and reduce their impact significantly. However, we observe slightly different results depending of the length of the atoms. In particular, dictionaries with shorter atom length are able to extract more high frequency features than those with larger atom length, but at the expense of leaving more energy at lower and middle frequencies. We have also explored the effect of the number of atoms. We find that once the number of atoms is sufficient large, above ∼2.5 times the length of the atoms, there is no appreciable improvement in the results.

Figure 2 displays the time-series of three illustrative examples of our sample of blip glitches reconstructed with a dictionary of 192 atoms, each with a length of 128 samples. Correspondingly, Fig. 3 shows the time-frequency diagram (spectrogram) of the same data. In all these examples (and in the whole set of glitches of our sample) the amplitude of the reconstruction is almost zero in the parts of the data stream that contain only background, and only the parts of the signals with larger amplitude are reconstructed. The examples in Fig. 2 reveal that the algorithm is able to reconstruct all blips present in the data. The subtraction residuals (bottom panels of Fig. 2) show that the impact of the glitches is significantly reduced in all cases. However, some non-negligible part of the glitch still remains in the data (see middle and right panels).

Let us now focus on the spectrograms shown in Fig. 3. The upper panels display the original data (after the whitening procedure) and the lower panels show the reconstructed spectrograms obtained when using a single dictionary of 192 atoms, each with a length of 128 samples. The three blips chosen to illustrate our procedure show several features that make them interesting cases of study. The left panel shows a blip with the typical tear-drop shape morphology. The middle panel displays a blip glitch with a strong contribution at high frequencies. Finally, a strong spectral line around ∼1400 Hz is well visible in the right panel and it is simultaneous to the occurrence of the blip glitch. The spectrograms of the reconstructed data confirm the analysis of the time-series plots. The power of the glitches is reduced for the low and middle frequencies (up to ∼500 Hz) while in other parts of the spectrogram both the signal and the background remain unaltered. This effect is clearly visible in the right plot of the lower panel of Fig. 3 which shows that the prominent spectral line is still present after the reconstruction. This is both a good and a bad feature of the method. On the one hand, it is a good result because, as the morphology of the line is totally different to that of the blip glitch, the dictionary does not reconstruct it; that would be a nice feature of the method in the case of a coincidence (both temporal and in frequency) of a blip glitch with an actual GW signal. On the other hand, it is a bad result because, as spectral lines are another source of noise, they require additional techniques to mitigate their impact.

The most obvious conclusion from the spectrograms is that the dictionary has difficulties in reducing the high-frequency content of the glitches. In the next subsection we discuss how to improve the performance at high frequencies and present quantitative estimates, using the SNR and W metrics, of our two deglitching procedures.

B. Combining dictionaries

We turn next to describe the results obtained when using a combination of different dictionaries to improve the results at high frequencies (above ∼500 Hz). This combination is a natural extension of our original algorithm based

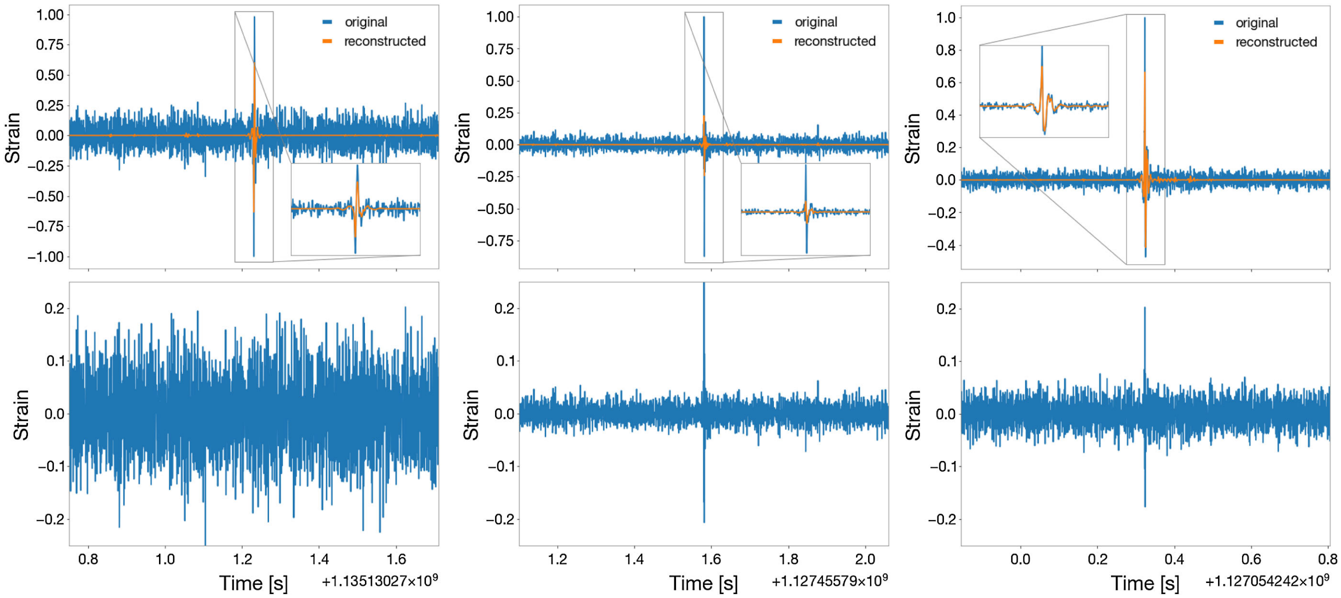

FIG. 2. Time-series plots of three illustrative examples of blip glitches, corresponding to numbers 7, 9 and 15 from the test set. The data have been band-passed between 100 Hz and 1 kHz to better show the morphology of the glitches. The upper panel shows the original signal (blue) with the reconstructed glitch (orange) superimposed using a dictionary of 192 atoms of 128 samples. The residual is shown in the bottom panel (Note that the interval of the vertical axes is smaller.).

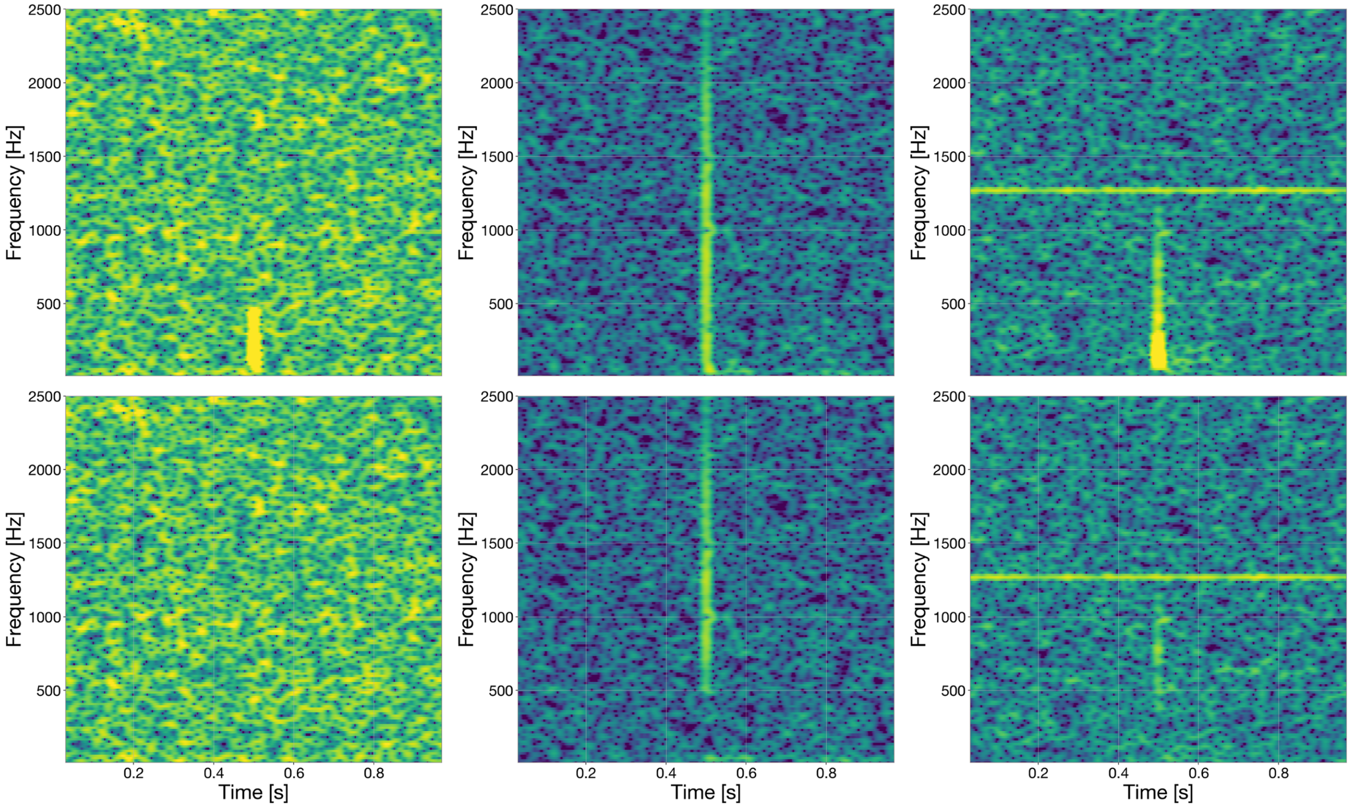

FIG. 3. Time-frequency diagram of the same three examples in Fig. 2. The original data and the residuals are represented in upper and bottom panel respectively.

on a single dictionary. Now the algorithm reads as follows: first we perform the reconstruction in the same way than before using the largest dictionary (i.e., 192 atoms of 128 samples length each). Then, a second dictionary is applied to the residual obtained from the first one. However, we do not use λ_{min} for this second dictionary. As our goal is to improve the results at high frequencies, we use a slightly lower value of λ, around 85% less. To keep the background from being affected due to this lower value of λ we restrict the reconstruction to a window which contains the most significant part of the blip. This window is determined by the reconstruction using the first dictionary, selecting the part of the signal which is not zero.

As an example we discuss results using a combination of two dictionaries, one formed by 192 atoms of 128 samples and another one comprising 40 atoms of only 16 samples. The results are presented in Figs. 4 and 5. The comparison of the time-series plots of Figs. 4 and 2 does not show an obvious improvement when two dictionaries are used instead of one. The inspection of Fig. 4 reveals that two of the peaks at the maximum of the glitch disappear in the left and right panels while part of the peak of the glitch in the center panel is also reduced. Comparing the spectrograms (i.e., Figs. 3 and 5) yields more meaningful information. One can observe that the high-frequency contribution that remains from the blip reconstruction with a single dictionary (up to ~1000 Hz) is further reduced by using a second, smaller dictionary. In addition, the background is not significantly perturbed. This also holds when the first dictionary is able to extract the blip glitch completely, as shown in the left panel of Fig. 5.

We have also analyzed the results for lower values of λ than the $0.85\lambda_{\text{min}}$ value used in this example. We find that the high-frequency component of the glitch can be reduced even more. However, at low and middle frequencies the background is affected and the spectrogram shows a significant reduction of power in the glitch there. Our tests indicate that a value of λ around 0.85λ min yields a good tradeoff.

Table I reports the two metrics, SNR and W, for all test cases shown in Figs. 2–5. The values of SNR and W for the original signals are shown in columns 2 and 5, respectively. The comparison using a single dictionary (columns 3 and 6) indicates that the dictionary is able to reduce the SNR significantly. The values of the whiteness increase sightly as expected. Note that as the data is whitened before the reconstruction, most of the original values of W are already close to one. For our procedure combining two dictionaries (columns 4 and 7 in Table I) the estimators show that in those cases where the first denoising does not reduce the glitch completely, the second dictionary is able to improve the results. In addition, in cases where the first

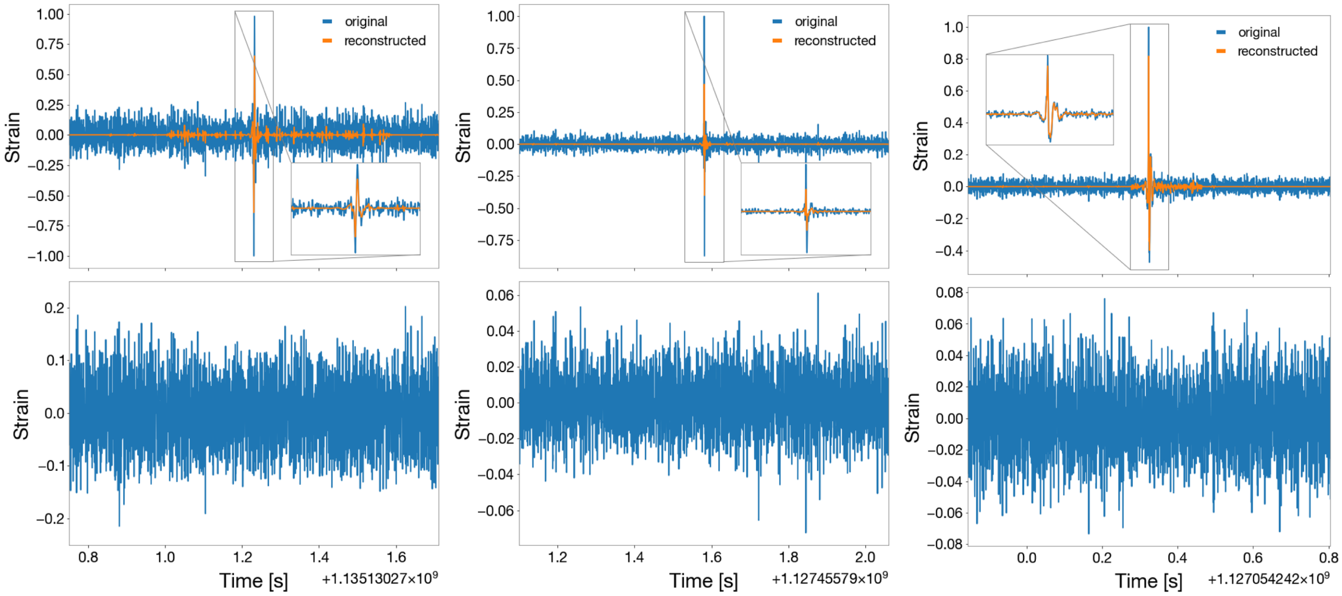

FIG. 4. Time-series plot of the same three examples of Fig. 2 when the reconstruction is done combining dictionaries. The data have been band-passed between 100 Hz and 1 kHz to better show the morphology of the glitches. The original data and the residuals are shown in the top and bottom panels, respectively.

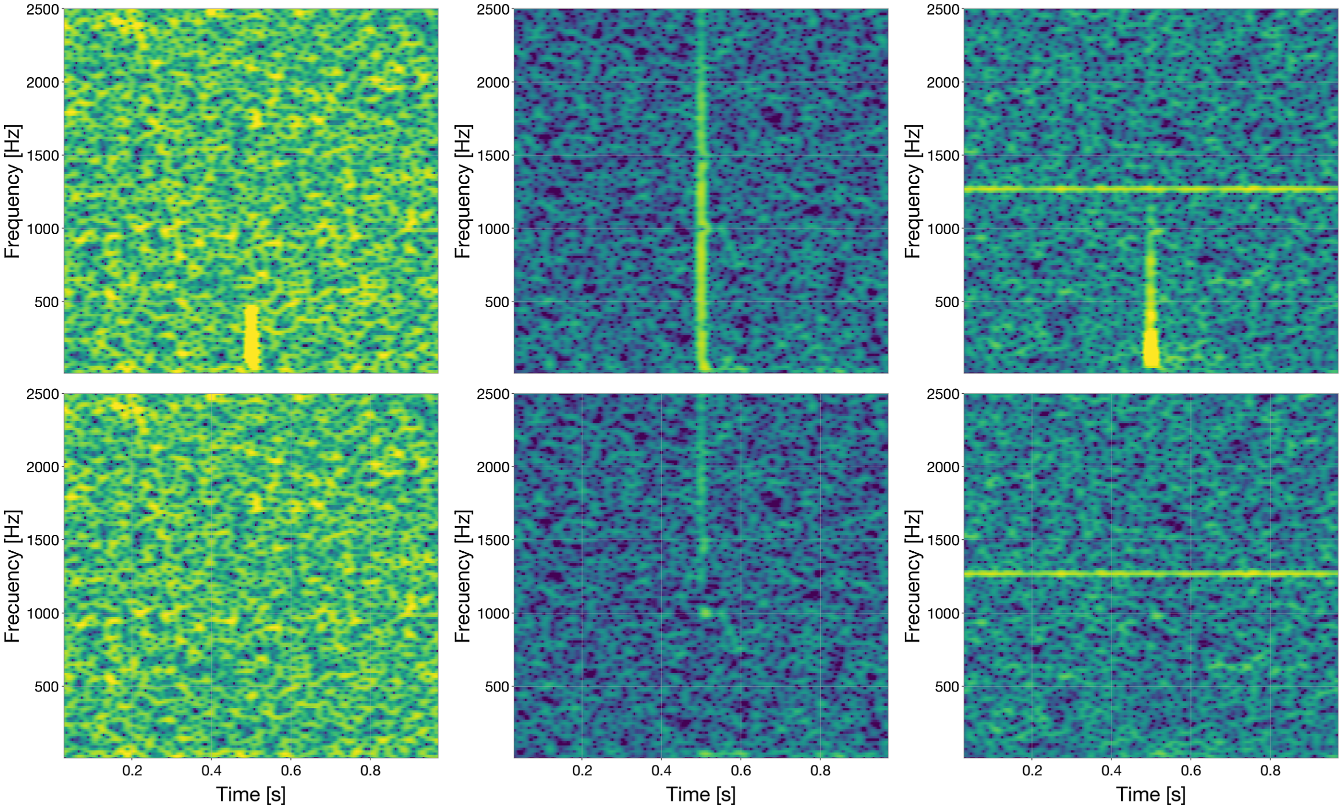

FIG. 5. Time-frequency diagram of the same three examples of Fig. 2 when using combined dictionaries.

dictionary already shows good performance and the SNR is reduced significantly, the addition of a second dictionary barely modifies the results. As a summary we conclude that, in general, glitch denoising with multiple dictionaries is a convenient strategy. For our test set of 16 blip glitches it reduces the SNR by a factor of ∼11 on average, while with a single dictionary the average reduction is ∼7.5, with a negligible increment in computational cost.

TABLE I. Quantitative assessment of our deglitching procedures. The columns report the values of our estimators, SNR and W, for the data containing the original noise transients (subindex "o") and for the residuals after deglitching, and both for a single dictionary (subindex "single") and for multiple dictionaries (subindex "multi").

Test No.	SNR_o	$\mathrm{SNR}_\mathrm{single}$	$\mathrm{SNR}_\mathrm{multi}$	W_o	$\mathrm{W}_\mathrm{single}$	$\mathrm{W}_\mathrm{multi}$
1	5.3	1.3	1.3	0.99	0.99	0.99
2	5.2	2.4	1.3	0.94	0.94	0.93
3	9.2	1.3	1.3	0.99	0.99	0.99
4	37.1	10.2	1.5	0.84	0.92	0.97
5	18.1	2.3	1.7	0.99	0.98	0.98
6	13.5	3.8	1.7	0.98	0.98	0.98
7	4.1	1.3	1.3	0.98	0.98	0.98
8	6.3	1.3	1.3	0.96	0.97	0.97
9	13.4	9.8	2.9	0.98	0.98	0.98
10	8.7	3.0	1.7	1.00	0.99	0.99
11	7.3	1.3	1.3	0.99	0.99	0.99
12	5.8	1.3	1.2	0.99	1.00	1.00
13	4.5	1.8	1.3	0.99	0.99	0.99
14	14.2	1.2	1.2	0.99	0.99	0.99
15	15.2	2.2	1.3	0.88	0.88	0.89
16	6.2	1.3	1.3	0.99	0.99	0.99

C. Deglitching of GW170817

In [4] the LIGO-Virgo Collaboration reported the first detection of GWs from a binary neutron star inspiral, GW170817. About 1.1s before the coalescence time of GW170817, a short instrumental noise transient appeared in the LIGO-Livingston detector (see Fig. 6, upper panel). The glitch was modeled with a time-frequency wavelet reconstruction and subtracted from the data, as shown in Fig. 2 of [4].

In this section we evaluate the use of learned dictionaries to deglitch the noise transient appearing in GW170817. The results are plotted in Fig. 6. We note that the spectrograms in this figure look different to those shown in the previous sections. This is because we use the routines of the Q-transform included in the GWpy libraries [43] in order to obtain similar plots to those reported in [4] to facilitate the comparison. It is worth stressing that the shape of the GW170817 glitch is not the same as that of the blip glitches we use to train our dictionaries. Therefore, these are not specifically tailored to deglitch the particular noise transient affecting GW170817. Nevertheless, this example still provides an excellent test to assess the capabilities of our trained dictionaries to reconstruct other types of glitches. In addition, the presence of the binary neutron star signal allows us to analyze if it is affected by the deglitching procedure.

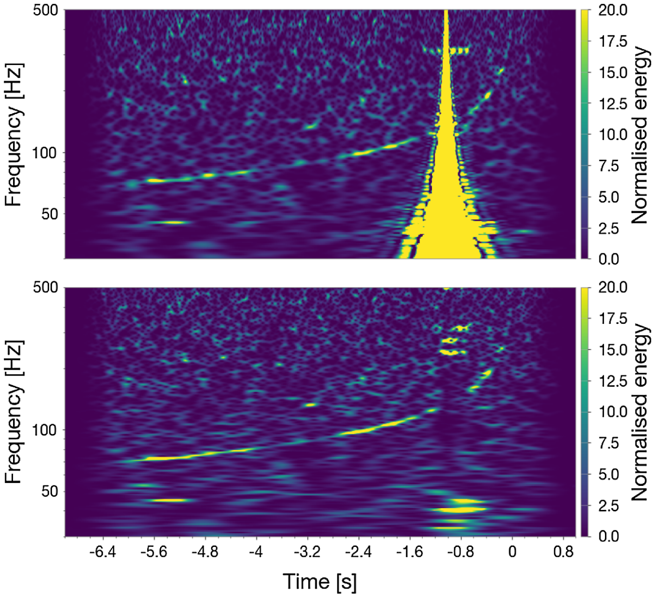

FIG. 6. Time-frequency diagram of 8 seconds of data corresponding to the GW170817 signal. The times shown are relative to August 17, 2017 12:41:04 UTC. The top panel shows the original data from LIGO-Livingston. The bottom panel displays the data after subtracting the glitch using a single blip-trained dictionary with 256 samples.

The bottom panel of Fig. 6 shows the results after applying one single dictionary of 256 samples, which is the dictionary that produces the best results in terms of reducing the contribution of both high and low frequencies. This is expected because the GW170817 glitch lasted significantly longer than the test cases discussed before. This figure shows that for the most part the glitch is removed from the data and, at the same time, the actual chirp signal from the inspiraling neutron stars behind the glitch is recovered almost completely. Nevertheless, part of the glitch at frequencies of ~400 Hz and below ~50 Hz remains visible after the first reconstruction. Subsequently applying a second dictionary composed by atoms of 16 samples does not significantly improve the results of the first dictionary in this example. This may be related to the fact that our trained dictionaries are not specifically customized to the morphological type of the GW170817 glitch. We note that these results could potentially improve if we trained our dictionaries with a larger set of glitch morphologies, instead of only using blips. We plan to investigate this possibility in the future.

V. SUMMARY

We have investigated the application of learned dictionaries to mitigate the effect of noise transients in the data of GW detectors. Although the data show the presence of many families of glitches, each with a different morphology and time-frequency shape, we have focused on "blip" glitches because they are the most common type of glitches found in the twin LIGO detectors and their waveforms are easy to identify. This paper is the first step in our ongoing program to automatically classify and subtract all families of glitches employing variational methods.

Our approach combines two different variational techniques based on the L_1 norm, namely the Rudin-Osher-Fatemi method [30] and a dictionary learning method [29].

We have randomly selected 100 blip glitches scattered in the data from advanced LIGO's O1 run. The data corresponds to a window of 1 s centred at the GPS time of each glitch as provided by Gravity Spy [16]. 85% of the glitches have been used to train the dictionary while the other 15% have been employed as examples to test the performance of the algorithm. The test set has included 16 blip glitches. In our approach we have incorporated a regularized ROF denoising step before the training step to obtain a smooth dictionary, which has proved to be more effective to model and subtract the glitches from the data.

The determination of a good value of the regularization parameter λ is not a trivial task. In this paper we have deviated from our previous works where the optimal value of λ could be determined by comparison with an analytical or numerical template [20,31–33]. To preserve the background as unaltered as possible, we find the first value of λ that only produces a reconstruction of the glitch and returns zeros for the surrounding background. This procedure has the advantage that λ is calculated only from the data and does not require any systematic search using templates. Our results have shown that this approach is valid to model and subtract most of the contribution of the blip glitches in all cases analyzed. They also have revealed, however, that the high-frequency component of the blips (above ∼500 Hz) is not completely removed. This issue has been ameliorated by using a combination of dictionaries with different atom length, one much shorter than the other. This seems to be the best strategy to mitigate the blips at all frequencies. We have also shown that our approach yields satisfactory results when applied to the GW170817 glitch [4] despite this transient noise feature is not of the blip glitch class. Learned dictionaries trained with a different set of glitches can remove the GW170817 glitch from the data without affecting the actual signal from the binary neutron star inspiral. Potentially, this result could be further improved by training the dictionaries with larger datasets accounting for other types of glitches.

Our results for GW170817 show that if the amplitude of the glitch is larger than the astrophysical signal, the latter is treated as part of the background. Even though this is a remarkable result it is unclear how our method performs in a superposition of a GW signal and a glitch with *comparable* amplitude and morphology. In our previous work [32] we analyzed a similar situation employing dictionaries to reconstruct numerically simulated signals injected in noise in coincidence, namely binary black hole signals and core-collapse supernova signals. We found that employing dictionaries individually learned for each type of signal allowed to separate the two. The possibility to accomplish glitch and GW signal separation using dictionaries will be explored elsewhere.

In order to eventually employ this approach in a low-latency pipeline in actual GW detectors, it would be necessary to improve the computational performance of the method. The current version of the code is implemented in MATLAB and not optimized. On average, the amount of time to process 1 s of data sampled at 8192 Hz is ∼1.5 s using a MacBook Pro with a 2,3 GHz Intel Core i5 processor and 16 Gb of RAM memory using 4 cores. We note, however, that the reconstruction can be trivially parallelized since each segment can be processed independently of the others. Therefore, there is still a considerable margin for improvement in execution time, developing an optimized and compiled version of the code. In the near future, we plan to extend the work initiated in [20] to account for other families of glitches and use classification algorithms as a previous step to the glitch subtraction procedure presented in this work.

ACKNOWLEDGMENTS

Work supported by the Spanish Agencia Estatal de Investigación (Grants No. PGC2018-095984-B-I00 and No. PGC2018-101119-B-I00), by the Generalitat Valenciana (PROMETEO/2019/071), by the European Union's Horizon 2020 RISE programme No. H2020-MSCA-RISE-2017 Grant No. FunFiCO-777740, and by the European Cooperation in Science and Technology through COST Action No. CA17137.

APPENDIX: PROPERTIES OF THE TEST SET

This Appendix summarizes the main properties and time-frequency characteristics of the 16 blips used to test our algorithms. The physical parameters are reported in Table II while Fig. 7 displays the time-frequency diagrams of all the 16 blips.

TABLE II. Physical parameters of the 16 blip glitches of the test set. Note that the SNR values reported in the table are those provided by Gravity Spy [16] and are different to the values of our SNR estimator.

Test No.	GPS time [s]	Peak frequency [Hz]	SNR	Amplitude	Central frequency [Hz]	Duration [s]	Bandwidth [Hz]	IFO
1	113 587 356 0.669	170.65	25.114	1.79×10^{-22}	294.23	0.375	521.87	H1
2	112 671 444 4.138	339.88	33.524	6.89×10^{-21}	809.47	0.188	1554.9	H1
3	112 7209 450.153	149.41	35.98	5.12×10^{-22}	300.04	0.281	536.07	L1
4	112 679 234 7.938	417.4	142.21	2.37×10^{-19}	1212.7	0.563	2361.5	L1

(Table continued)

TABLE II. *(Continued)*

Test No.	GPS time [s]	Peak frequency [Hz]	SNR	Amplitude	Central frequency [Hz]	Duration [s]	Bandwidth [Hz]	IFO
5	113 586 605 7.468	211.48	58.114	4.34×10^{-22}	1926.7	0.625	3798.3	H1
6	113 240 024 7.362	324.75	40.023	2.52×10^{-21}	1262.8	0.281	2457.2	L1
7	113 513 027 1.231	211.48	18.767	1.77×10^{-22}	2421.1	0.248	4737.3	L1
8	113 281 005 0.112	137.71	29.714	2.75×10^{-22}	298.75	0.375	512.83	L1
9	112 745 579 1.581	1166	40.429	1.47×10^{-20}	1563.5	0.375	3063.1	H1
10	113 277 974 1.205	402.44	27.316	1.98×10^{-20}	2391.7	0.263	4698.8	L1
11	113 512 999 0.751	211.48	30.26	2.83×10^{-22}	720.44	0.375	1372.6	L1
12	113 515 384 5.843	262.06	26.046	5.26×10^{-22}	440.05	0.156	824.97	L1
13	113 194 993 9.534	324.75	19.041	5.65×10^{-22}	1934.1	0.094	3783.6	H1
14	112 679 539 9.374	183.5	50.77	7.09×10^{-22}	444.39	0.313	824.77	L1
15	112 705 424 2.325	149.41	53.247	4.29×10^{-22}	1232.6	0.313	2386.5	H1
16	113 585 488 5.481	137.71	30.468	2.57×10^{-22}	194.54	0.625	333.95	H1

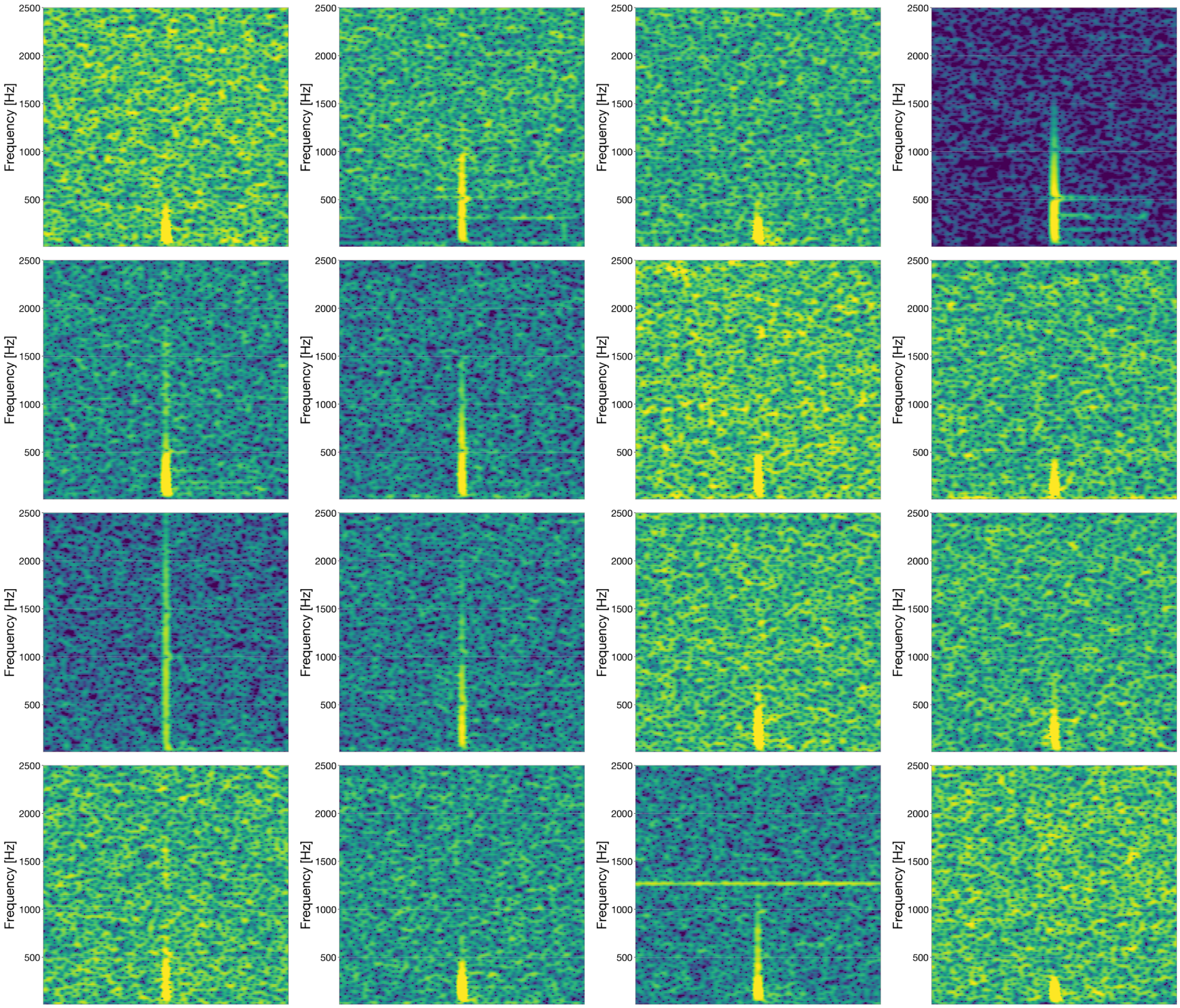

FIG. 7. Time-frequency diagrams of all the blip glitches used as a test set.

[1] J. Aasi, B. P. Abbott, R. Abbott, T. Abbott, M. R. Abernathy, K. Ackley, C. Adams, T. Adams, P. Addesso *et al.* (LIGO Scientific Collaboration), Classical Quantum Gravity **32,** 115012 (2015).

[2] F. Acernese *et al.* (Virgo Collaboration), Classical Quantum Gravity **32,** 024001 (2015).

[3] B. P. Abbott, R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, R. Adhikari *et al.* (LIGO Scientific and Virgo Collaborations), Phys. Rev. X **9,** 031040 (2019).

[4] B. P. Abbott, R. Abbott, T. D. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, V. B. Adya *et al.*, Phys. Rev. Lett. **119,** 161101 (2017).

[5] B. P. Abbott, R. Abbott, T. D. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, V. B. Adya *et al.*, Astrophys. J. Lett. **848,** L12 (2017).

[6] A. Albert, M. André, M. Anghinolfi, M. Ardid, J. J. Aubert, J. Aublin, T. Avgitas, B. Baret, J. Barrios-Martí, S. Basa *et al.*, Astrophys. J. Lett. **850,** L35 (2017).

[7] GW Candidate Event Database, https://gracedb.ligo.org.

[8] Y. Aso, Y. Michimura, K. Somiya, M. Ando, O. Miyakawa, T. Sekiguchi, D. Tatsumi, and H. Yamamoto, Phys. Rev. D **88,** 043007 (2013).

[9] D. V. Martynov, E. D. Hall, B. P. Abbott, R. Abbott, T. D. Abbott, C. Adams, R. X. Adhikari *et al.*, Phys. Rev. D **93,** 112004 (2016).

[10] B. P. Abbott, R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, V. B. Adya *et al.* (LIGO Scientific and Virgo Collaborations), Classical Quantum Gravity **37,** 055002 (2020).

[11] J. R. Smith, T. Abbott, E. Hirose, N. Leroy, D. MacLeod, J. McIver, P. Saulson, and P. Shawhan, Classical Quantum Gravity **28,** 235005 (2011).

[12] B. P. Abbott, R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, V. B. Adya *et al.* (LIGO Scientific and Virgo Collaborations), arXiv:1908.11170.

[13] B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari *et al.*, Classical Quantum Gravity **35,** 065010 (2018).

[14] J. Powell, D. Trifirò, E. Cuoco, I. S. Heng, and M. Cavaglià, Classical Quantum Gravity **32,** 215012 (2015).

[15] J. Powell, A. Torres-Forné, R. Lynch, D. Trifirò, E. Cuoco, M. Cavaglià, I. S. Heng, and J. A. Font, Classical Quantum Gravity **34,** 034002 (2017).

[16] M. Zevin, S. Coughlin, S. Bahaadini, E. Besler, N. Rohani, S. Allen, M. Cabero, K. Crowston, A. K. Katsaggelos, S. L. Larson *et al.*, Classical Quantum Gravity **34,** 064003 (2017).

[17] N. Mukund, S. Abraham, S. Kandhasamy, S. Mitra, and N. S. Philip, Phys. Rev. D **95,** 104059 (2017).

[18] D. George and E. A. Huerta, Phys. Lett. B **778,** 64 (2018).

[19] M. Razzano and E. Cuoco, Classical Quantum Gravity **35,** 095016 (2018).

[20] M. Llorens-Monteagudo, A. Torres-Forné, J. A. Font, and A. Marquina, Classical Quantum Gravity **36,** 075005 (2019).

[21] S. Coughlin, S. Bahaadini, N. Rohani, M. Zevin, O. Patane, M. Harandi, C. Jackson, V. Noroozi, S. Allen, J. Areeda *et al.*, Phys. Rev. D **99,** 082002 (2019).

[22] R. E. Colgan, K. R. Corley, Y. Lau, I. Bartos, J. N. Wright, Z. Marka, and S. Marka, Phys. Rev. D **101,** 102003 (2020).

[23] C. Pankow, K. Chatziioannou, E. A. Chase, T. B. Littenberg, M. Evans, J. McIver, N. J. Cornish, C.-J. Haster, J. Kanner, V. Raymond *et al.*, Phys. Rev. D **98,** 084016 (2018).

[24] B. Zackay, T. Venumadhav, J. Roulet, L. Dai, and M. Zaldarriaga, arXiv:1908.05644 [Phys. Rev. D (to be published)].

[25] T. Venumadhav, B. Zackay, J. Roulet, L. Dai, and M. Zaldarriaga, Phys. Rev. D **100,** 023011 (2019).

[26] W. Wei and E. A. Huerta, Phys. Lett. B **800,** 135081 (2020).

[27] M. Cabero, A. Lundgren, A. H. Nitz, T. Dent, D. Barker, E. Goetz, J. S. Kissel, L. K. Nuttall, P. Schale, R. Schofield *et al.*, Classical Quantum Gravity **36,** 155010 (2019).

[28] A. H. Nitz, Classical Quantum Gravity **35,** 035016 (2018).

[29] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, in *Proceedings of the 26th Annual International Conference on Machine Learning* (ACM, New York, 2009), pp. 689–696.

[30] L. I. Rudin, S. Osher, and E. Fatemi, Physica (Amsterdam) **60D,** 259 (1992).

[31] A. Torres, A. Marquina, J. A. Font, and J. M. Ibáñez, Phys. Rev. D **90,** 084029 (2014).

[32] A. Torres-Forné, A. Marquina, J. A. Font, and J. M. Ibáñez, Phys. Rev. D **94,** 124040 (2016).

[33] A. Torres-Forné, E. Cuoco, A. Marquina, J. A. Font, and J. M. Ibáñez, Phys. Rev. D **98,** 084013 (2018).

[34] B. A. Olshausen and D. J. Field, Vision Res. **37,** 3311 (1997).

[35] M. Aharon, M. Elad, and A. Bruckstein, IEEE Trans. Signal Process. **54,** 4311 (2006).

[36] S. S. Chen, D. L. Donoho, and M. A. Saunders, SIAM Rev. **43,** 129 (2001).

[37] R. Tibshirani, J. R. Stat. Soc. Ser. B **58,** 267 (1996).

[38] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Found. Trends® Mach. Learn. **3,** 1 (2011).

[39] M. Elad and M. Aharon, IEEE Trans. Image Process. **15,** 3736 (2006).

[40] A. Chambolle, in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition* (Springer, New York, 2005), pp. 136–152.

[41] E. Cuoco, G. Calamai, L. Fabbroni, G. Losurdo, M. Mazzoni, R. Stanga, and F. Vetrano, Classical Quantum Gravity **18,** 1727 (2001).

[42] E. Cuoco, G. Losurdo, G. Calamai, L. Fabbroni, M. Mazzoni, R. Stanga, G. Guidi, and F. Vetrano, Phys. Rev. D **64,** 122002 (2001).

[43] D. Macleod, A. L. Urban, S. Coughlin, T. Massinger, M. Pitkin, P. Altin, J. Areeda, E. Quintero, and K. Leinweber, *GWpy: Python Package for Studying Data from Gravitational-Wave Detectors* (2019).