# Convolutional neural network to distinguish glitches from minute-long gravitational wave transients

Vincent Boudart[*]

*STAR Institute, Bâtiment B5, Université de Liège, Sart Tilman B4000 Liège, Belgium*

(Received 12 October 2022; accepted 19 December 2022; published 9 January 2023)

Gravitational wave bursts are transient signals distinct from compact binary mergers that arise from a wide variety of astrophysical phenomena. Because most of these phenomena are poorly modeled, the use of traditional search methods such as matched filtering is excluded. Bursts include short ($<10$ s) and long (from 10 s to a few hundred seconds) duration signals for which the detection is constrained by environmental and instrumental transient noises called glitches. Glitches contaminate burst searches, reducing the amount of useful data and limiting the sensitivity of current algorithms. It is therefore of primordial importance to locate and distinguish them from potential burst signals. We propose training a convolutional neural network to detect glitches in the time-frequency space of the cross-correlated LIGO noise. We show that our network is retrieving more than 95% of the glitches while being trained on only a subset of the existing glitch classes, thus highlighting the sensitivity of the network to completely new glitch classes.

DOI: 10.1103/PhysRevD.107.024007

## I. INTRODUCTION

Gravitational waves (GWs) were detected on September 14, 2015 [1] by the Advanced LIGO [2] detectors, thereby revealing the collision of two black holes for the first time. Since then, the Advanced LIGO and the Advanced Virgo [3] detectors have observed more than 90 compact binary coalescence (CBC) events [4], among which are black hole–neutron star [5] and binary neutron star collisions [6]. In light of the planned sensitivity improvement of the Advanced LIGO and Advanced Virgo detectors, a new family of gravitational wave sources, known as unmodeled GW transients or bursts, is a prime target candidate for the next observing run. Bursts include a wide range of astrophysical phenomena for which accurate waveforms are not accessible. The computational resources required to build a template bank covering a wide range of complex and highly turbulent events prevents us from using matched filtering methods such as those in CBC searches [7]. Some of the expected progenitors of gravitational wave transients are supernovae [8], fallback accretion events [9], accretion-disk instabilities [10], nonaxisymmetric deformations in magnetars [11], and gamma-ray bursts [12]. Two classes of bursts are identified: short ($<10$ s) and long (from 10 s to a few hundred seconds). In this paper, we present a new machine learning tool that complements our previous work [13] and discriminates transient noises happening in the detectors from long-duration burst signals.

The main approach to detecting burst events while making minimal assumptions on the targeted signals relies on the excess-of-power method. It consists in searching for excess of power in the time-frequency space of single or multiple detector data, i.e., to find narrow time-evolving frequency curves. This problem has already been tackled by different groups who built the current generation of pipelines—namely, PySTAMPAS [14], cocoA [15], the two different versions of STAMP-AS, Zebragard and Lonetrack [16,17], the long-duration configuration of coherent WaveBurst [18] and X-SphRad [19].

One of the main hindrances in burst searches is glitches. Glitches are transient noises caused by instrumental or environmental sources [20,21] that appear in the detector data in large quantities. Several families of glitches have been reported [22], showing different time-frequency morphologies. Glitches limit the sensitivity of the searches and can hinder GW detections. Therefore, all of the aforementioned pipelines deal with glitches in either pre- or postprocessing steps. In a previous work [13], we trained a neural network with chirp signals having random parameters and showed that our methodology can be used to detect minute-long GW transients. However, it can also recover glitches fairly, and a visual inspection is needed to discriminate them from chirp signals. This work aims to remove the false alarms caused by glitches through a convolutional neural network.

Convolutional neural networks (CNNs) were recently used in burst detection [23]. Skliris *et al.* [23] built a one-dimensional CNN to detect generic short-duration signals

[*] vboudart@uliege.be

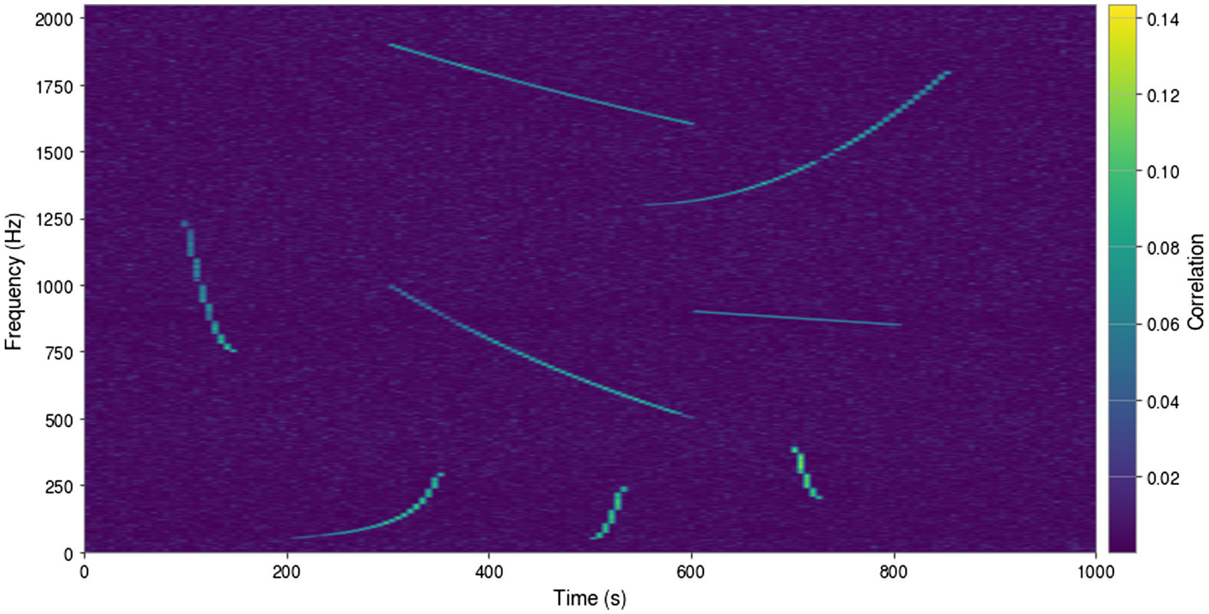

FIG. 1. Examples of chirp signals.

from the strain data of the LIGO and Virgo detectors. CNNs have shown promising results in the identification and classification of GW bursts from supernovae [24,25], in the detection of binary black hole mergers [26] as well as long-duration transients from isolated neutron stars [27], and as early-alert systems for binary neutron star collisions [28]. CNNs are widely used for pattern recognition [13,29] and classification tasks [30–32]. Their powerful capability to identify shapes and structures has led to the definition of generative adversarial networks [33], thus allowing new samples to be generated by learning the underlying distribution of the original data.

In Sec. II, we describe how glitches have been selected to constitute the training set and how we highlight them in the cross-correlated time-frequency (TF) maps. Details about the architecture of our classifier and the training method are given in Sec. III. We then show the results of the training in Sec. IV. Section V is dedicated to large-scale tests comparable to the analyses conducted during burst searches. Future prospects and conclusions are given in Sec. VI.

## II. METHODOLOGY

Our search for minute-long bursts is based on the excess-of-power method [34]. We make use of correlated spectrograms, also referred to as TF maps, as described in Ref. [13]. In order to distinguish glitches from possible burst signals, we will train a neural network to identify them in the spectrograms. As both can be present in a single TF map, we need to consider the following cases: (1) a glitch is present in the map, (2) a burst signal is present in the map, (3) both of them show up in the spectrogram, and (4) neither of them show up in the spectrogram. Accordingly we will build four different datasets to include all the possible scenarios in the training phase.

The fourth scenario consists in building a dataset with background TF maps. The data from Hanford (H1) and Livingston (L1) from the first half of the third observing run (O3a) are whitened [35] prior to be correlated. Using time slides [36], we then generate 10000 spectrograms with a time resolution of 6 s and a frequency resolution of 2 Hz. As the TF maps span 1000 s and 2048 Hz, their size is 166 × 1025. Since we aim to apply our classifier on ALBUS's output, the size of the TF maps is chosen to be identical to [13].

### A. Chirp generation

A methodology to recognize minute-long burst signals using machine learning techniques with very few assumptions was proposed in our previous work [13]. This approach consists in using the `SciPy` library [37] to generate chirp signals in the time domain with random parameters, covering the entire time-frequency parameter space. Figure 1 shows some examples of generated chirps. As has been shown [13], this allows one to train a neural network with no prior assumption of the targeted signals while confidently identifying minute-long burst models. Chirps are injected into noise with nine levels of visibility, defined as

$$V = \sum_{i,j} (S_{ij} - N_{ij}), \tag{1}$$

where $N_{ij}$ is a noise-only spectrogram and $S_{ij}$ refers to the same spectrogram in which a signal has been injected. The sum is carried over all the pixels $(i, j)$ in the map. The definition of the visibility is particularly useful to ensure that chirps are visible in the TF maps, thus preventing the network from being fooled during the training phase. The visibility can also be seen as a measure of the anomalousness of the input TF maps. We choose nine intensity levels in order to cover a quite large intensity range, as seen in Fig. 15 in the Appendix. We use this intensity criterion to build our second dataset, which contains 10000 samples.

### B. Glitch selection

During the second observing run, glitches happened at a rate of roughly one per minute in the detectors [38]. Although it amounts to a considerable volume of contaminated data, the glitches barely show up in cross-correlated spectrograms. Indeed, both glitches have to fall into overlapping time bins while showing a sufficiently high signal-to-noise ratio (SNR) and sharing some frequency bandwidth. Even if these conditions greatly reduce the amount of glitches that contaminate our search, several thousand glitches can be found out of a couple million TF maps generated during the background searches.

To constitute our dataset with glitches, we need a way to inject several glitch classes into time-frequency maps. However, the only tool that is currently available to produce realistic glitches can only generate *blip* glitches [39]. Blips are one of the 23 classes that have been characterized by Gravity Spy [22,40]. They have a frequency between 0 and 256 Hz [41,42] which would limit the detection bandwidth of the classifier if used exclusively in our dataset. Therefore, we have to rely on the glitches detected thus far to constitute the training set. We thus select glitches that have been recorded by Gravity Spy during O3a [43]. We load the data around the Global Positioning System time of the chosen glitch in each single detector (H1 and L1) and shift them so that they fall into the same time bin. In this way, we maximize the probability of finding cross-correlated glitches that appear clearly in the TF maps. Moreover, glitches showing higher SNRs do not always lead to stronger cross-correlated signals in the TF maps. To circumvent these problems, we choose seven glitch classes with SNRs ranging from 20 to 10000 in both the Hanford and Livingston data. This will ensure some variability in the results of the cross-correlation. Table I summarizes the useful information.

TABLE I. Information about the glitches selected from H1 and L1.

| | |
|---|---|
| Glitch classes | Blip, low frequency burst, scattered light, tomte, whistle, extremely loud, koi fish |
| SNR ranges | 20–30, 30–40, 40–50, 50–100, 100–150, 150–200, 200–300, 300–500, 500–10000 |
| Number per range | 30 (if possible) |
| Injection time | 50–950 s |
| Total | H1: 1110 L1: 1260 |

The total number of selected glitches is 1110 for H1 and 1260 for L1. We randomly choose one glitch from each detector and build the resulting time-frequency map. We reproduce this procedure 50000 times. To evaluate whether the cross-correlation of the chosen glitches has led to a visible glitch in the output spectrogram, we employ ALBUS, the neural network dedicated to burst detection [13]. We showed that ALBUS can recover glitches as well as chirp signals. We use its output map to introduce a score quantifying the anomalousness present in the original spectrogram, which is called the anomaly score (AS). This score is defined as

$$\mathrm{AS} = \sum_{i,j} O_{i,j} \quad \text{if } O_{i,j} > 0.5 \max(O), \tag{2}$$

where $O$ is the ALBUS output map and $i$ and $j$ indicate the time and frequency dimensions. The anomaly score can be thought of the sum over the pixels remaining after an intensity cut to the output map is applied. This threshold has been chosen to exclude all the values close to zero, as they are quite numerous given the size of the TF maps and can have an impact on the final anomaly score. The anomaly score can also be used to rank detected signals as seen in Fig. 2, where an extended glitch shows a higher score than a glitch that has a small frequency range.

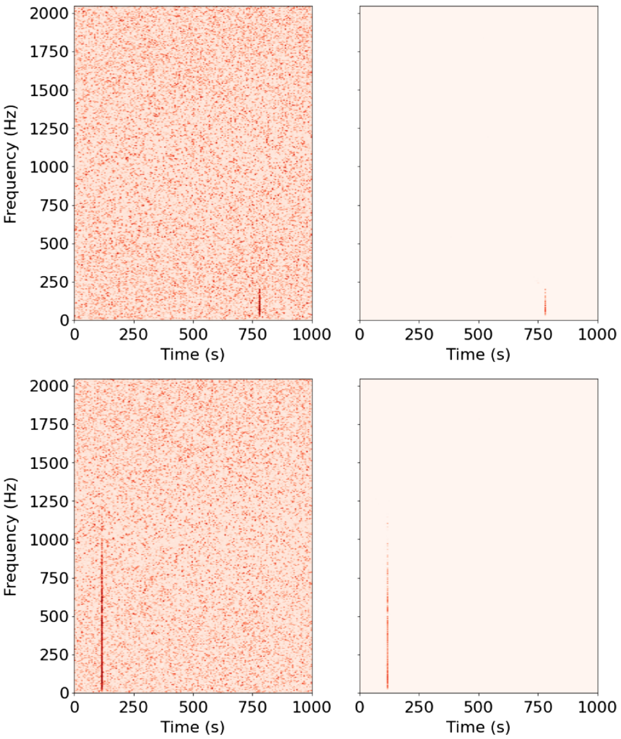

FIG. 2. Examples of correlated glitches with different anomaly scores. The left panels show the generated spectrogram, while the right panels show the output of ALBUS. The top and bottom glitches have anomaly scores of 21 and 144, respectively.

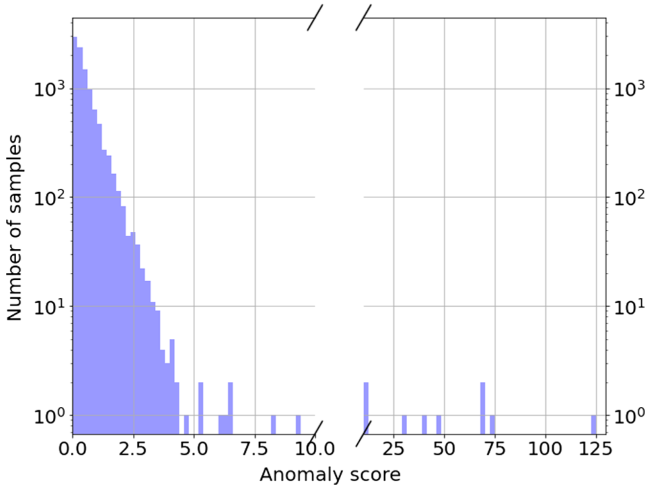

FIG. 3. Histogram of the anomaly scores for 10000 background TF maps.

After visual inspection, background maps without any glitch have a maximum score of around 6.5, as seen in Fig. 3, where 10000 images have been processed. All 15 background images with scores above 8 show a correlated glitch. We thus set the threshold to confirm the presence of a correlated glitch at 8 in order to leave a sufficient margin between high-noise-level TF maps and those containing glitches. After applying this threshold to our 50000 spectrograms, we end up with only 4744 maps showing correlated glitches. The dataset has been drastically reduced but it is still sufficient to achieve a well-behaved training.

### C. Combined dataset

The procedure to generate spectrograms containing a chirp and a glitch is very similar to the method described in the previous subsection. We generate 45000 spectrograms with the glitches selected in Table I. Then we inject chirp signals with nine levels of visibility, as in Sec. II A. Once the signal has been added to the map, we process the latter with ALBUS.

At this stage, we cannot rely on the anomaly score as it is defined. The chirp signals will also be recovered and contribute to the anomaly score of the map, which can hide the presence of a glitch. However, as we know where the chirp is injected, we can discard the corresponding pixels in the output map. This is done by masking the pixels corresponding to the footprint of the injected chirp. Therefore, the anomaly score is still relevant to assess whether or not a glitch is present in the maps. Out of the 45000 maps, 6068 actually pass the threshold and contain a correlated glitch and an injected chirp. This thorough check for glitches is important in light of the training approach explained in the next section.

Figure 4 shows an example of a spectrogram containing a chirp and a correlated glitch. The anomaly score of the

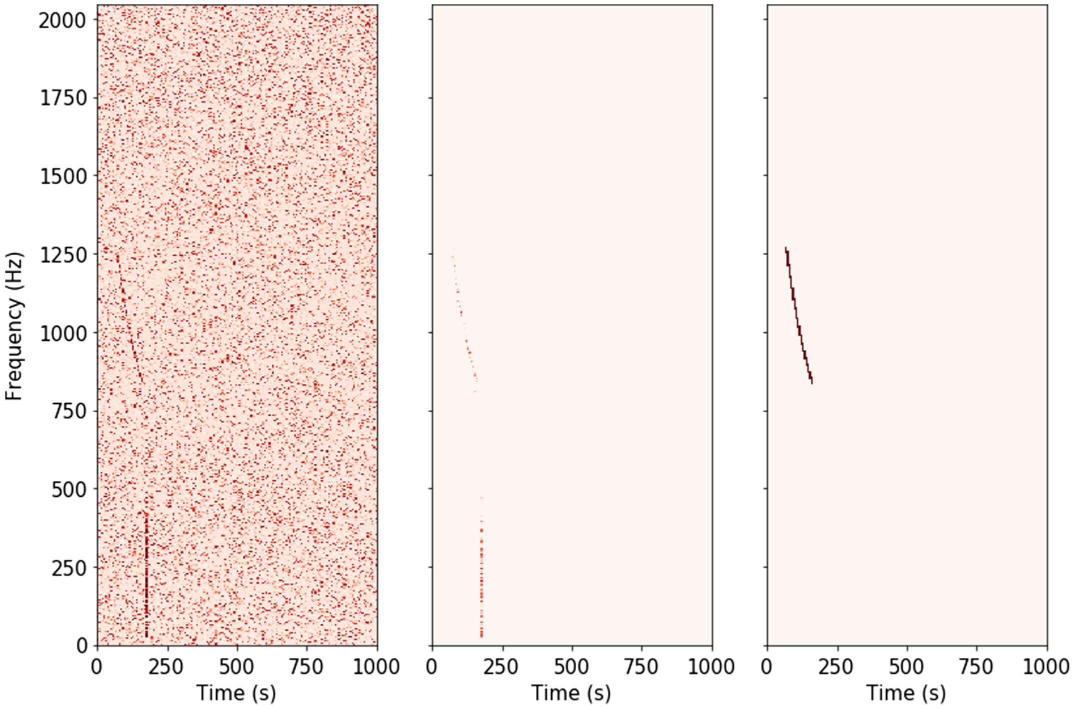

FIG. 4. Example of a TF map showing a glitch and a chirp signal. The left and center panels, respectively, illustrate the generated spectrogram and the output of ALBUS. The right panel corresponds to the chirp mask that is used to cancel out the contribution of the chirp in the estimated anomaly score.

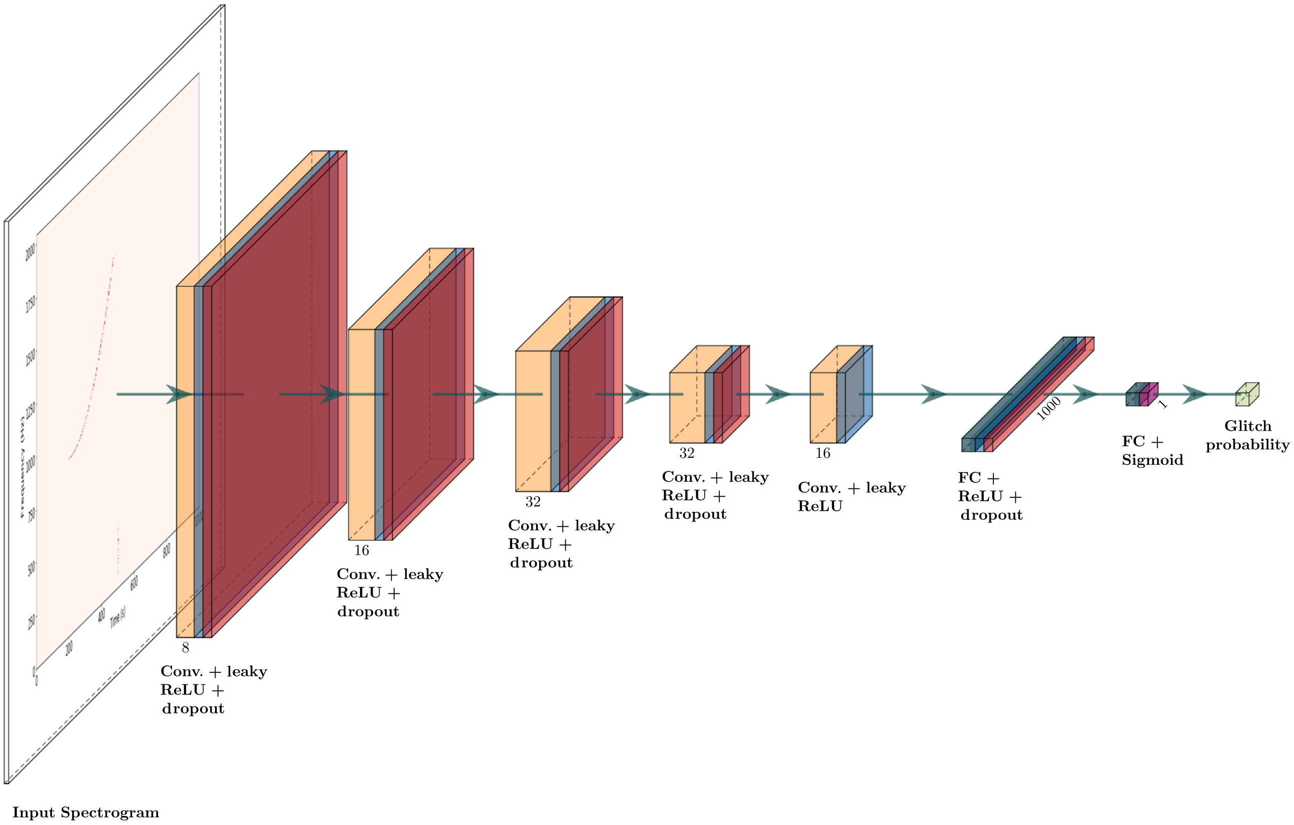

FIG. 5. Architecture of the CNN. Conv., convolution; FC, fully connected layers; ReLU, rectified linear unit.

output map is 49.6, while it drops to 34.5 when the chirp pixels are masked out.

## III. MACHINE LEARNING

In this work, we use a CNN to assess whether a glitch is present in the time-frequency maps. For this, we feed the output of ALBUS [13] to a CNN, predicting a glitch probability. The full architecture can be seen in Fig. 5. The network is composed of two parts. The first part is fully convolutional and acts as a feature extractor. Then a fully connected network uses these features to evaluate a glitch probability. The sigmoid activation function is used to obtain an output value between 0 and 1. The hyperparameters of the network have been chosen via trial and error. We add dropout [44] to every convolution layer and the first dense layer with a probability of 30%. Table II shows an exhaustive list of the parameters used across all layers.

TABLE II. Hyperparameters used in the architecture of our classifier.

| | Nb of filters | Kernel size | Stride | Padding |
|---|---|---|---|---|
| Conv. 1 | 8 | $7 \times 7$ | $2 \times 2$ | $0 \times 0$ |
| Conv. 2 | 16 | $7 \times 7$ | $1 \times 1$ | $0 \times 0$ |
| Conv. 3 | 32 | $5 \times 5$ | $2 \times 2$ | $0 \times 0$ |
| Conv. 4 | 32 | $5 \times 5$ | $1 \times 1$ | $0 \times 0$ |
| Conv. 5 | 16 | $3 \times 3$ | $2 \times 2$ | $0 \times 0$ |
| FC 1 | 29280 | … | … | … |
| FC 2 | 1000 | … | … | … |

The training procedure is straightforward. Every TF map is passed through the network with a glitch label, as summarized in Table III. The binary cross entropy (BCE) loss is applied between the predicted and real label:

$$L = \mathrm{BCE}(L_g, P_g), \tag{3}$$

with BCE defined as

TABLE III. Labels used for the training of the classifier.

| TF maps | Glitch label |
|---|---|
| Background | 0 |
| Chirp | 0 |
| Glitch | 1 |
| Combined | 1 |

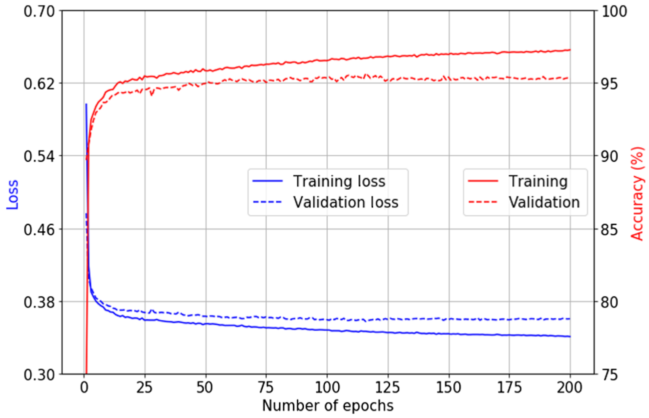

FIG. 6. Loss and accuracy of the network for a training of 200 epochs.

$$\mathrm{BCE}(x, y) = y \log(x) + (1 - y) \log(1 - x), \qquad (4)$$

where $L_g$ stands for the glitch label and $P_g$ is the predicted glitch probability.

## IV. RESULTS

### A. Training

We select 4000 TF maps in each category, amounting to 16000 images for our dataset. A validation set of 20% is used throughout the training. We use the ADAMAX optimizer, a variant of ADAM [45], with a weight decay of $10^{-5}$ and a learning rate of $3 \times 10^{-5}$. The batch size is set at 32. The evolution of the loss and the accuracy of predicting the glitch labels are shown in Fig. 6. Both the training and validation losses behave smoothly during the training. We stop the training when the validation loss starts to rise again, indicating that the network starts to overfit the data. At the same time, the accuracy reaches a plateau and no further progress is observed. After 200 epochs, the validation accuracy reaches 95.5%. The training time is roughly 2 h on a Tesla P100 GPU (16 GB).

### B. Classification

To assess the identification of glitches, we run the classifier on the remaining images of each class—namely, 6000 for the background and chirp class, 744 for the glitch class, and 2068 for the combined class. The confusion matrix is shown in Fig. 7. The threshold value to decide whether a TF map contains a glitch is chosen to be 0.5. Glitches appearing in the data are found with an accuracy above 95.55%, while background and chirp images are correctly identified in at least 90.17% of the cases. Note that the false-alarm rate for background images is very low, with roughly 0.33% of the TF maps misclassified.

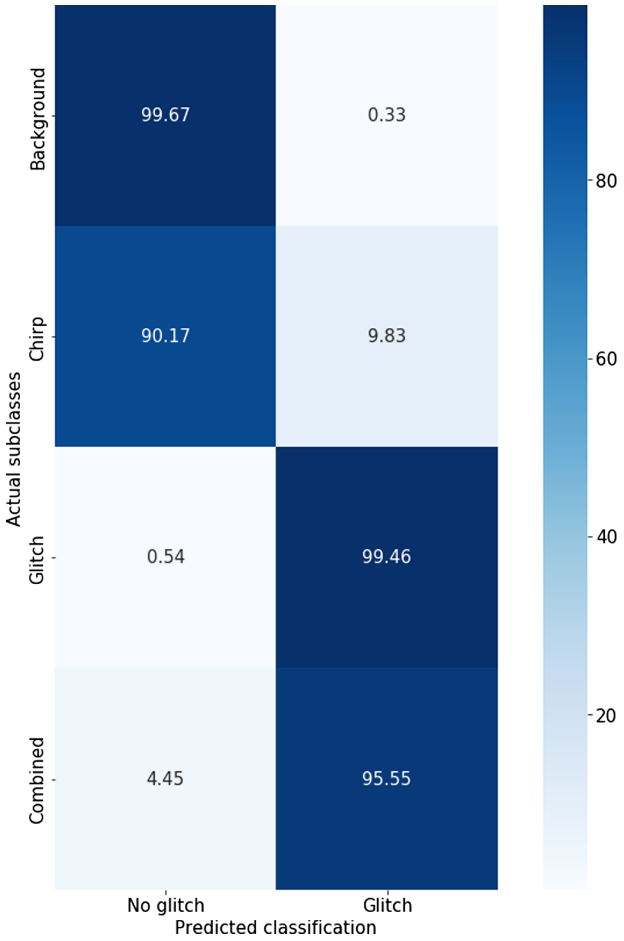

FIG. 7. Confusion matrix for the glitch label. The test has been conducted on 2812 TF maps showing glitches (744 glitch images and 2068 combined images) and 12000 that do not include any glitch (6000 background images and 6000 chirp images).

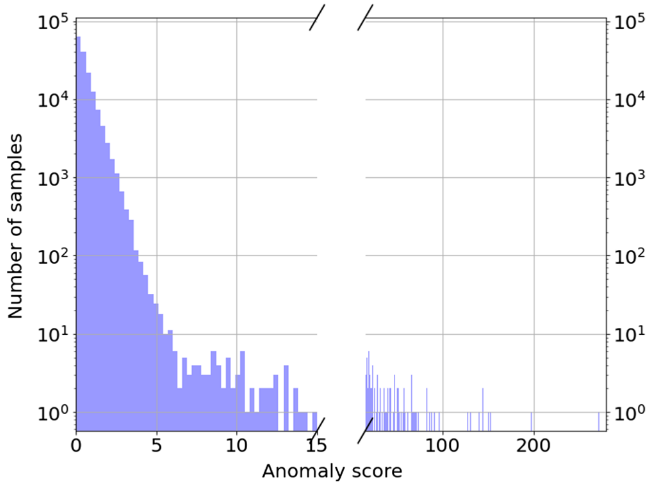

FIG. 8. Histogram of the anomaly scores for five years of background.

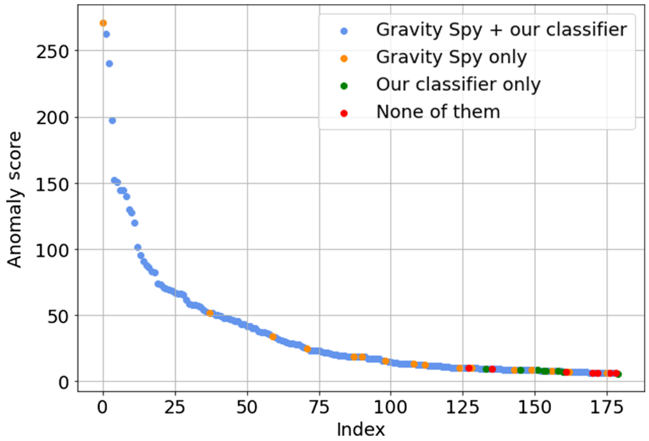

FIG. 9. Classification of the 180 highest background candidates.

## V. LARGE-SCALE TESTS

### A. Background analysis

To test whether the trained CNN can be used to reduce the false-alarm rate of ALBUS during a real search, we simulated a five-year background search, accounting for 157772 time-frequency maps to process. The background is produced via time slides [36] with real data from Hanford and Livingston from the O3a run. Every image is passed to ALBUS to filter the background components. Its output is then used to evaluate the anomaly score of the map and finally passed through the glitch classifier. If the glitch probability is above 0.5, the TF map is classified as containing a glitch.

The background distribution of the anomaly scores is shown in Fig. 8. As most of the images show a small anomaly score, they will not limit our sensitivity to burst signals. However, some background candidates get a score above 6 and should be examined.

Among the highest candidates, we expect to find a majority of cross-correlated glitches. To compare the classifier with state-of-the-art glitch retrieval procedures, we use Gravity Spy [22]. For every image, we check to see whether Gravity Spy has recorded a glitch in either the Hanford or Livingston data at that time. All the TF maps showing an anomaly score above 6 (180 in total) have been analyzed, and their classification as images containing a glitch is shown in Fig. 9. Gravity Spy retrieves 165 glitches while our classifier identifies 157 of them, having 149 glitches in common. Gravity Spy cancels out candidates with a high anomaly score, but some of them (16 in total) are missed by our CNN. The output of ALBUS for some of these TF maps is shown in Fig. 10. The glitches shown look like classical glitches, although they present a higher minimal frequency than those appearing in Figs. 2 and 4. A probable explanation is that our classifier is sensitive to the bandwidth of the signals. By cross-correlating only seven classes of glitches, we have limited the variability in the resulting TF map, somehow indirectly impacting the detection capability of the network.

On the other hand, the classifier recognizes eight glitches at low anomaly scores (the green dots in Fig. 9) for which Gravity Spy does not detect anything. Since Gravity Spy takes Omicron [46] triggers as input, the latter might not have produced triggers for these eight

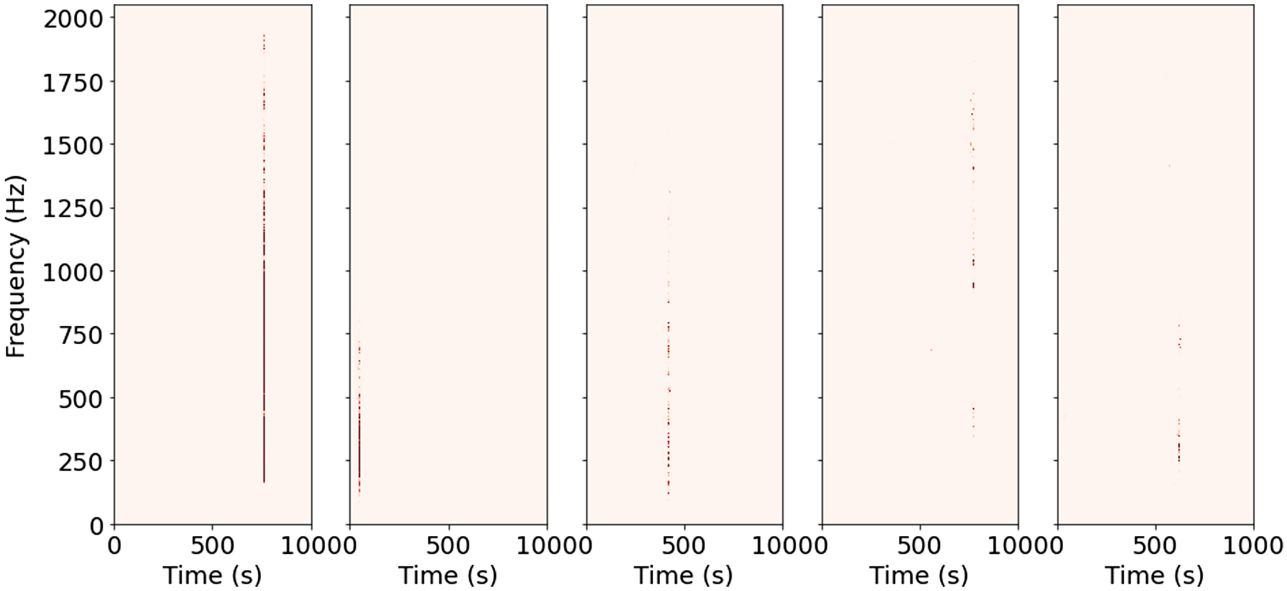

FIG. 10. Outputs of ALBUS for TF maps in which Gravity Spy has identified a glitch while our classifier has not detected any.

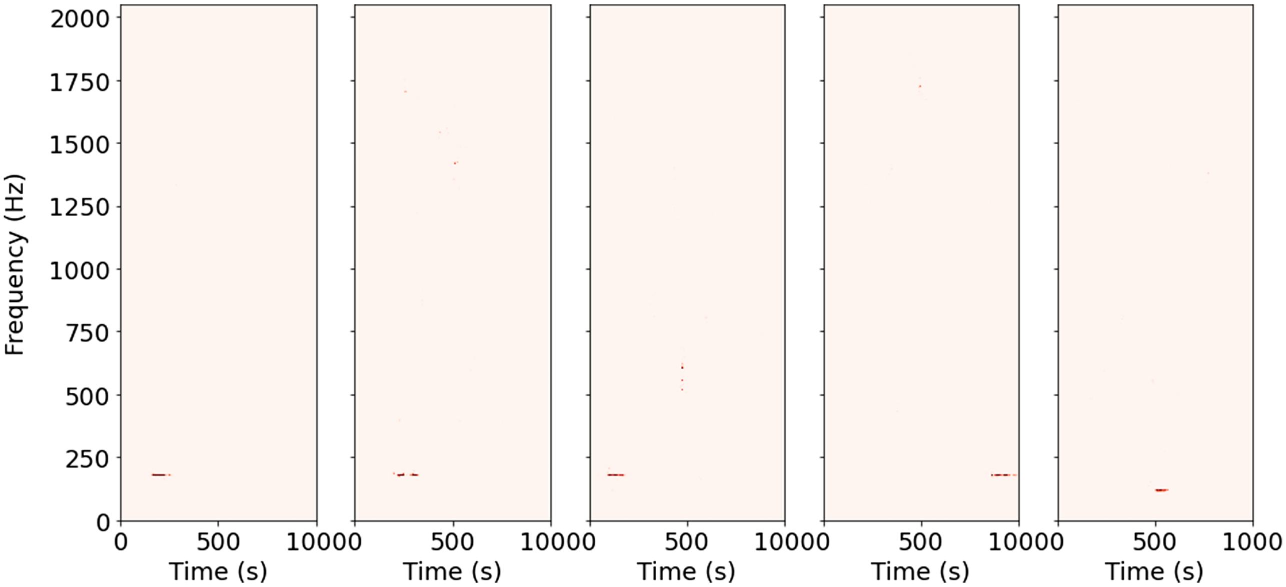

FIG. 11. Outputs of ALBUS for TF maps in which our classifier has identified a glitch while Gravity Spy has not detected any.

events. Figure 11 shows the output of ALBUS, i.e., the input of the classifier, for five of them. The patterns are narrowband and last 100–150 s. As these artifacts appear as monofrequency lines, they might be related to the power line at 60 Hz in the U.S. and its harmonics [38]. Figure 12 shows the spectrogram before and after the whitening for one of the examples in Fig. 11 and reveals that the first harmonic (120 Hz) barely appears in the data. Therefore, the whitening procedure could not clean that power line.

### B. Burst signal analysis

To complete the tests carried out on background images, we applied our CNN to four expected types of long-duration burst signals across 22 hrss (root-sum-squared strain amplitude) intensities. In order to claim a detection, the output of ALBUS should contain an anomaly, but it cannot be a glitch. The second condition is met when the glitch probability (GP) is lower than 0.5. To validate the first condition, the AS of the output map should be higher than the anomaly score obtained for the background images. In Fig. 9, the highest anomaly score that is not identified as a glitch by Gravity Spy for a background map is 10.24. The two thresholds used for the analysis are therefore

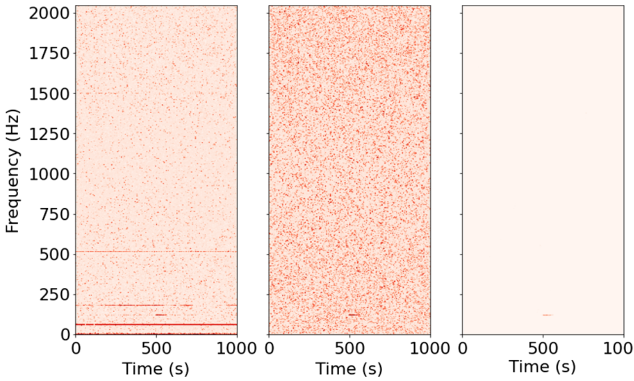

FIG. 12. Example of artifact detected by our classifier. The left and central panels, respectively, show the generated spectrogram before and after the whitening procedure, while the right panel shows the output of ALBUS.

$$\mathrm{GP} < 0.5, \tag{5}$$

$$\mathrm{AS} > 10.24. \tag{6}$$

Figure 13 shows the efficiency curves for four different waveforms in two different scenarios. Every dot is the estimation over 200 injections performed at the same hrss intensity.

The detection efficiency for the long-duration waveforms is highly dependent on the shape of the footprint left in the TF maps. Figure 14 shows the pattern left by the four selected waveforms. The detection efficiency for the magnetar model is very similar whether or not the glitch probability is used, meaning that the classifier does not recognize it as a glitch. This is not the case for the three other models. Indeed, our classifier identifies events of that kind as glitches most of the time, which is certainly due to their steep behavior. Moreover, these sorts of chirps might not be abundant enough in the data since the chirp generation parameters have been randomized. The classifier might therefore consider only the steep part to classify them as glitches.

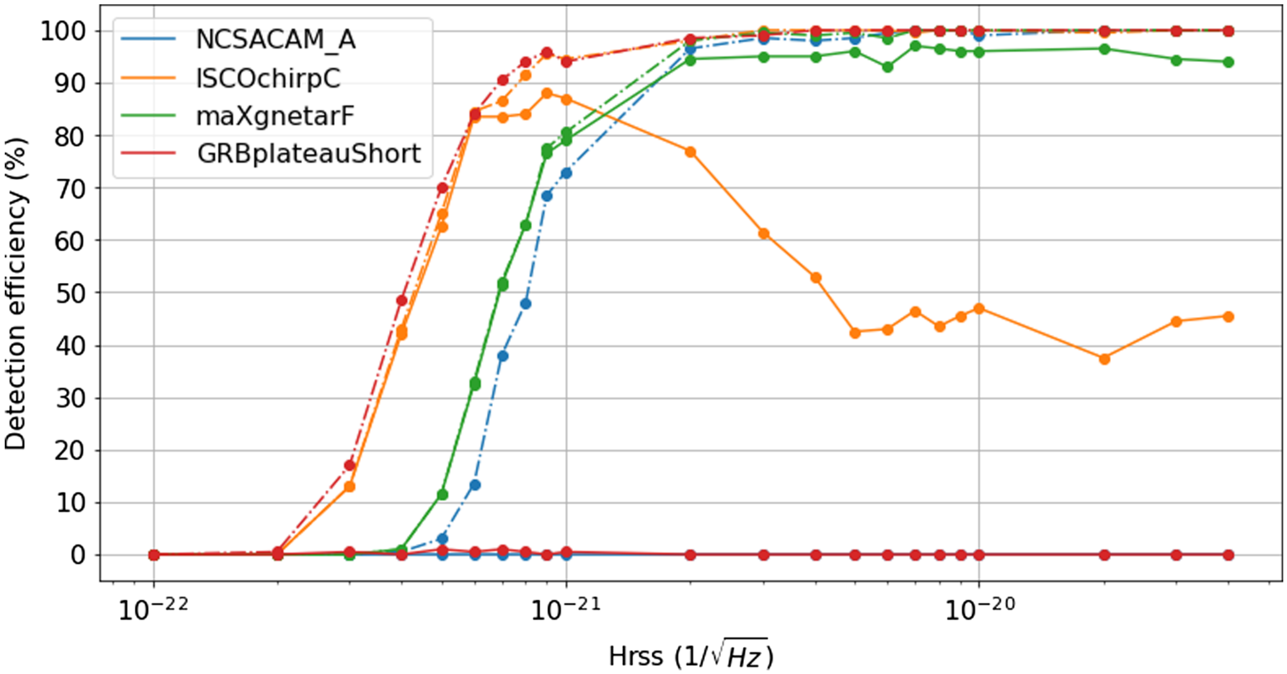

FIG. 13. Detection efficiency for long-duration waveforms (GRBplateauShort [47], ISCOchirpC [48], maXgnetarF [49], NCSACAM-A [50]). The dash-dotted curves refer to a scenario where the only threshold is the anomaly score, while the glitch probability is also used in the case of the continuous lines.

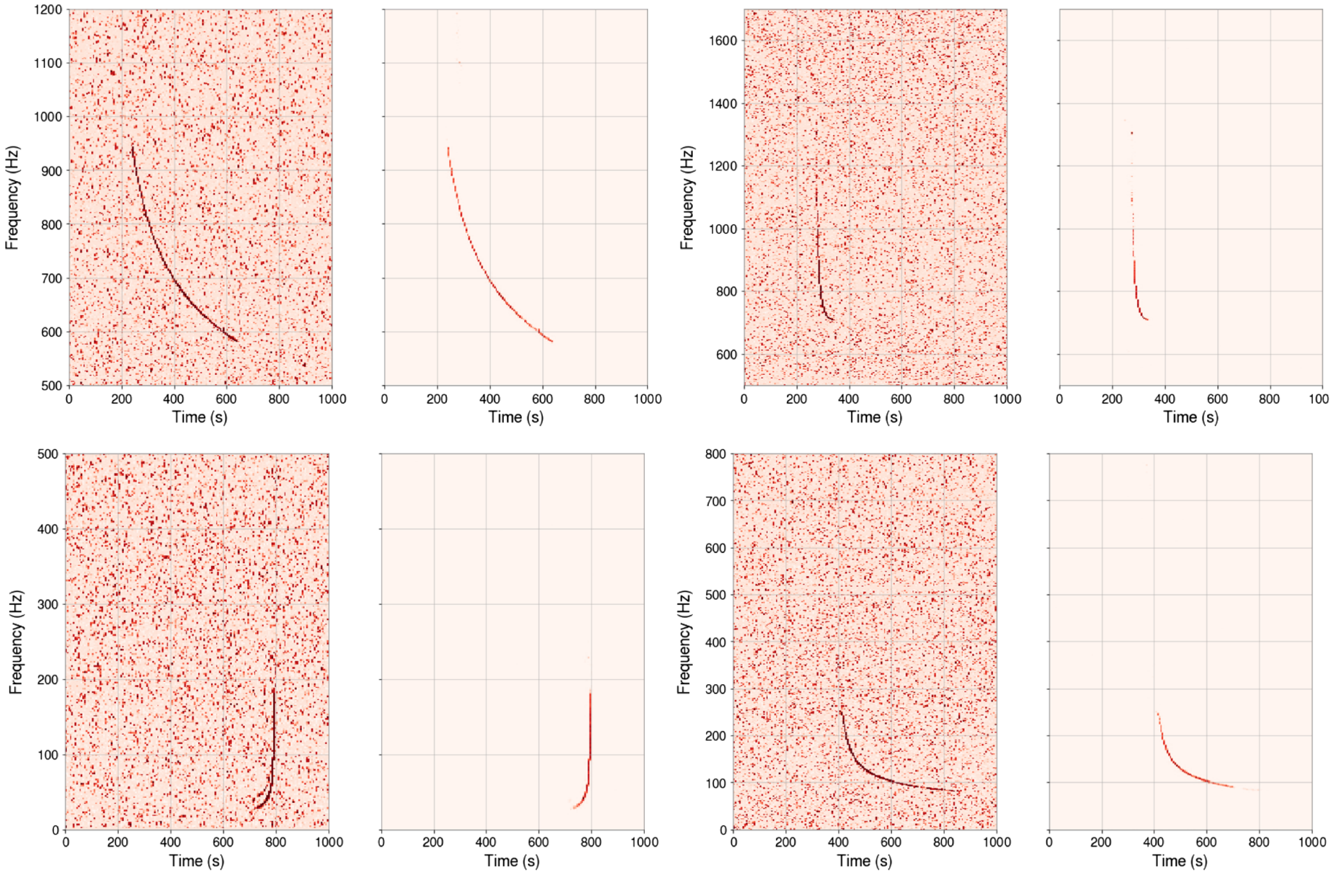

FIG. 14. Examples of detection performance on long-duration waveforms (top left panel: maXgnetar-D [49]; top right panel: ISCOchirp-C [48]; bottom left panel: NCSACAM-A [50]; bottom right panel: GRBplateau [47]). The left image in each panel is the input TF map, and the right panel shows the output of ALBUS.

## VI. DISCUSSION AND CONCLUSION

The anomaly score has been defined as a statistics to detect and rank signals in the output map of ALBUS. It can also be used as the unique detection threshold of our pipeline, showing encouraging results. Gravity Spy [22] could then be used to remove the false alarms due to glitches.

In this paper, we have applied a convolutional neural network to the identification of detector glitches in the time-frequency space of the cross-correlated LIGO noise. The training has been carried out with both glitches and chirping signals to help the network learn their distinct morphologies. The network recognizes more than 95% of the glitches, while it has a low false-alarm rate on random chirping signals. The performance of the classifier can be improved by adding more glitch classes to the training data, increasing accordingly the variability in their cross-correlation output. Indeed, we only select seven glitch classes in this work, limiting the bandwidth diversity in the data.

In the same way, chirp data have to be adapted to improve the performances on long-duration models showing a steep pattern in the TF space. The efficiency of our network can be improved by either overpopulating rapidly chirping signals in the data or by training directly on a subset of long-duration waveforms. Future works will contribute to the improvement of the classifier introduced in this paper.

## ACKNOWLEDGMENTS

The author thanks Grégory Baltus for useful discussions and comments. V. B. is supported by a Gravitational Wave Science (GWAS) grant funded by the French Community of Belgium. This material is based upon work supported by NSF’s LIGO Laboratory which is a major facility fully funded by the National Science Foundation. The author is grateful for computational resources provided by the LIGO Laboratory and supported by the National Science Foundation Grants No. PHY-0757058 and No. PHY-0823459.

## APPENDIX: VISIBILITY LEVELS

Figure 15 shows the nine levels of visibility that have been used to inject chirp signals into the chirp and combined training sets.

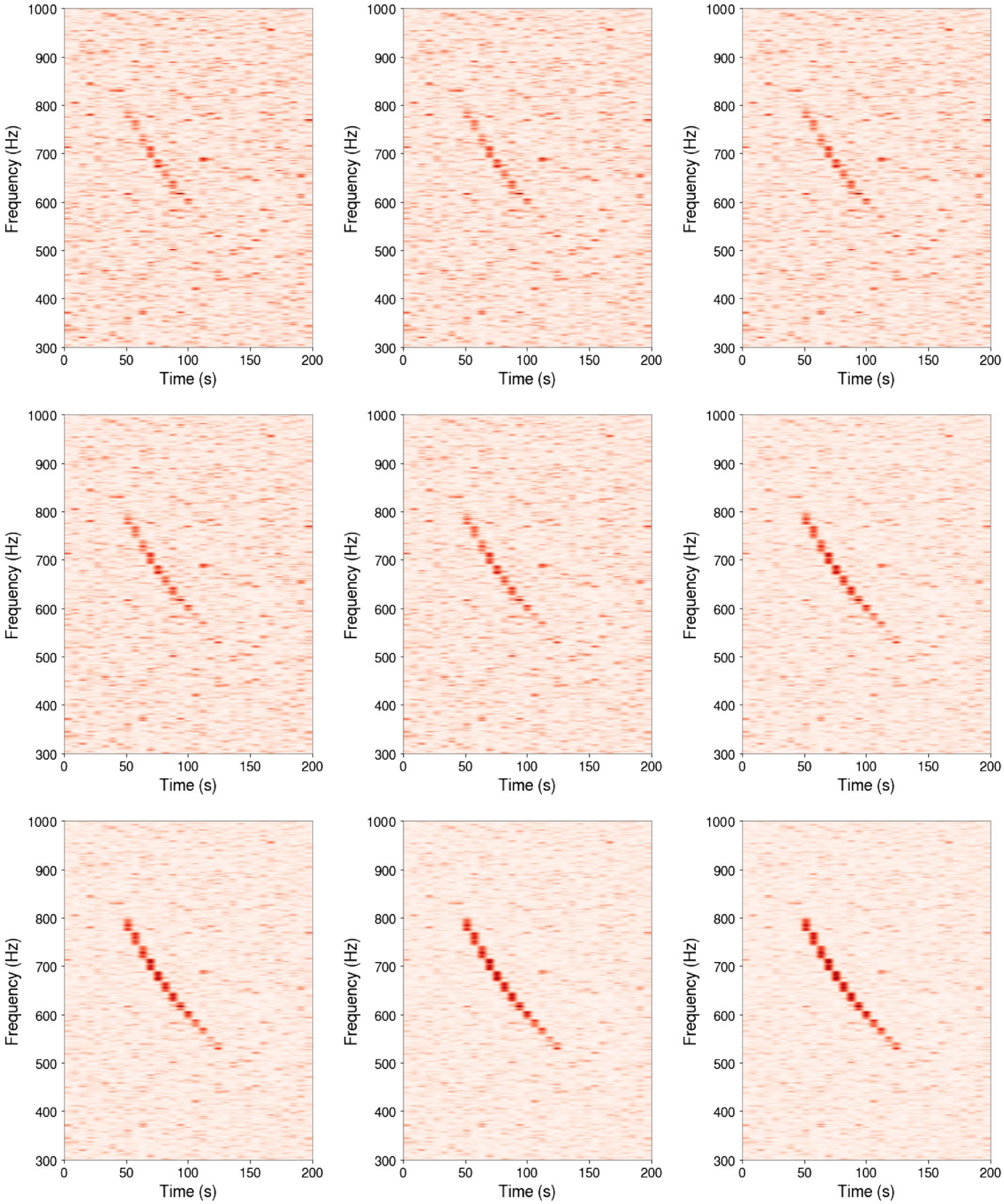

FIG. 15. Visibility levels used in this work shown through a unique injected chirp. The values are, from top left to bottom right, 12, 14, 16, 18, 20, 30, 40, 50, and 60.

[1] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Phys. Rev. Lett. **116**, 061102 (2016).

[2] J. Aasi *et al.*, Classical Quantum Gravity **32**, 074001 (2015).

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

[4] R. Abbott *et al.* (LIGO Scientific, Virgo, and KAGRA Collaborations), arXiv:2111.03606.

[5] R. Abbott *et al.* (LIGO Scientific, Virgo, and KAGRA Collaborations), Astrophys. J. Lett. **915**, L5 (2021).

[6] B. P. Abbott *et al.*, Phys. Rev. Lett. **119**, 161101 (2017).

[7] C. Cutler and E. E. Flanagan, Phys. Rev. D **49**, 2658 (1994).

[8] M. H. P. M. van Putten, A. Levinson, F. Frontera, C. Guidorzi, L. Amati, and M. Della Valle, Eur. Phys. J. Plus **134**, 537 (2019).

[9] A. Piro and E. Thrane, Astrophys. J. **761**, 63 (2012).

[10] M. H. P. M. van Putten, Phys. Rev. Lett. **87**, 091101 (2001).

[11] S. Dall'Osso, B. Giacomazzo, R. Perna, and L. Stella, Astrophys. J. **798**, 25 (2015).

[12] A. Corsi and P. Meszaros, Astrophys. J. **702**, 1171 (2009).

[13] V. Boudart and M. Fays, Phys. Rev. D **105**, 083007 (2022).

[14] A. Macquet, M. A. Bizouard, N. Christensen, and M. Coughlin, Phys. Rev. D **104**, 102005 (2021).

[15] R. Coyne, A. Corsi, and B. J. Owen, Phys. Rev. D **93**, 104059 (2016).

[16] E. Thrane and M. Coughlin, Phys. Rev. Lett. **115**, 181102 (2015).

[17] T. Prestegard, Ph.D. thesis, University of Minnesota, 2016, https://hdl.handle.net/11299/182183.

[18] S. Klimenko, G. Vedovato, M. Drago, F. Salemi, V. Tiwari, G. A. Prodi, C. Lazzaro, K. Ackley, S. Tiwari, C. F. Da Silva *et al.*, Phys. Rev. D **93**, 042004 (2016).

[19] M. Fays, Ph.D. thesis, Cardiff University, 2017, http://orca.cardiff.ac.uk/id/eprint/110245.

[20] J. Aasi *et al.*, Classical Quantum Gravity **32**, 115012 (2015).

[21] B. P. Abbott *et al.*, Classical Quantum Gravity **33**, 134001 (2016).

[22] M. Zevin *et al.*, Classical Quantum Gravity **34**, 064003 (2017).

[23] V. Skliris, M. Norman, and P. Sutton, arXiv:2009.14611.

[24] M. Lopez, I. Di Palma, M. Drago, P. Cerda-Duran, and F. Ricci, Phys. Rev. D **103**, 063011 (2021).

[25] A. Iess, E. Cuoco, F. Morawski, and J. Powell, Mach. Learn. **1**, 025014 (2020).

[26] A. Menendez-Vazquez, M. Kolstein, M. Martinez, and L. M. Mir, Phys. Rev. D **103**, 062004 (2021).

[27] A. L. Miller, P. Astone, S. D'Antonio, S. Frasca, G. Intini, I. La Rosa, P. Leaci, S. Mastrogiovanni, F. Muciaccia, A. Mitidis *et al.*, Phys. Rev. D **100**, 062005 (2019).

[28] G. Baltus, J. Janquart, M. Lopez, A. Reza, S. Caudill, and J. R. Cudell, Phys. Rev. D **103**, 102003 (2021).

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016* (IEEE, Los Alamitos, CA, 2016), pp. 779–788.

[30] C. Szegedy, L. Wei, J. Yangqing, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, 2015* (IEEE, New York, 2015), pp. 1–9.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, in *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS '12), Vol. 1, Lake Tahoe, NV, 2012*, edited by E. Pereira, C. J. C. Burges, and L. Bottou (Curran Associates, Red Hook, NY, 2012), p. 1097–1105.

[32] M. Tan and Q. Le, Proc. Mach. Learn. Res. **97**, 6105.

[33] I. J. Goodfellow *et al.*, in *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS '14), Vol. 2, Montreal, 2014*, edited by Z. Ghahramani, M. Welling, and C. Cortes (MIT Press, Cambridge, MA, 2014), pp. 2672–2680.

[34] E. E. Flanagan and S. A. Hughes, New J. Phys. **7**, 204 (2005).

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

[36] M. Was, M. A. Bizouard, V. Brisson, F. Cavalier, M. Davier, P. Hello, N. Leroy, F. Robinet, and M. Vavoulidis, Classical Quantum Gravity **27**, 015005 (2010).

[37] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, Nat. Methods **17**, 261 (2020).

[38] D. Davis *et al.*, Classical Quantum Gravity **38**, 135014 (2021).

[39] M. Lopez, V. Boudart, K. Buijsman, A. Reza, and S. Caudill, Phys. Rev. D **106**, 023027 (2022).

[40] S. Soni, C. P. L. Berry, S. B. Coughlin, M. Harandi, C. B. Jackson, K. Crowston, C. Østerlund, O. Patane, A. K. Katsaggelos, L. Trouille *et al.*, Classical Quantum Gravity **38**, 195016 (2021).

[41] B. P. Abbott *et al.*, Classical Quantum Gravity **33**, 134001 (2016).

[42] B. P. Abbott *et al.*, Classical Quantum Gravity **35**, 065010 (2018).

[43] J. Glanzer, S. Banagiri, S. B. Coughlin, S. Soni, M. Zevin, C. P. L. Berry, O. Patane, S. Bahaadini, N. Rohani, K. Crowston *et al.*, arXiv:2208.12849.

[44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, J. Mach. Learn. Res. **15**, 1929 (2014).

[45] D. Kingma and J. Ba, in *Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, 2015*, edited by Y. Bengio and Y. LeCun (Association for Computing Machinery, New York, 2015).

[46] F. Robinet, N. Arnaud, N. Leroy, A. Lundgren, D. Macleod, and J. McIver, SoftwareX **12**, 100620 (2020).

[47] A. Corsi and P. Mészáros, Astrophys. J. **702**, 1171 (2009).

[48] M. H. P. M. Van Putten, Astrophys. J. **819**, 169 (2016).

[49] S. Dall'Osso, B. Giacomazzo, R. Perna, and L. Stella, Astrophys. J. **798**, 25 (2015).

[50] E. A. Huerta, P. Kumar, B. Agarwal, D. George, H.-Y. Schive, H. P. Pfeiffer, R. Haas, W. Ren, T. Chu, M. Boyle *et al.*, Phys. Rev. D **95**, 024038 (2017).