

Gravitational-wave signal recognition of LIGO data by deep learning

He Wang¹, Shichao Wu², Zhoujian Cao^{2,*}, Xiaolin Liu², and Jian-Yang Zhu¹

¹*Department of Physics, Beijing Normal University, Beijing 100875, China*

²*Department of Astronomy, Beijing Normal University, Beijing 100875, China*



(Received 1 October 2019; accepted 8 April 2020; published 1 May 2020)

The deep learning method has developed very fast as a tool for data analysis in recent years. Moreover, as a technique, it is quite promising as a way to analyze gravitational-wave detection data. Multiple works in the literature have already used deep learning to process simulated gravitational-wave data. In this paper, we apply deep learning to LIGO data. In order to improve the weak signal recognition, we design a new structure of the convolutional neural network (CNN). The key feature of our new CNN structure is the sensing layer. This layer mimics matched filtering but is different from the usual matched-filtering technique. Usually, the matched-filtering technique uses a full template bank to match the data. However, our sensing layer only uses tens of waveforms. Our new convolutional neural network admits comparable accuracy and efficiency of signal recognition compared to other deep learning works published in the literature. Based on our new CNN, we can clearly recognize the 11 confirmed gravitational-wave events included in O1 and O2. In addition, we find about 2000 gravitational-wave triggers in O1 data.

DOI: [10.1103/PhysRevD.101.104003](https://doi.org/10.1103/PhysRevD.101.104003)

I. INTRODUCTION

Gravitational waves (GW) are an important prediction of Einstein's general theory of relativity, which was published a century ago. Gravitational-wave observations of coalescing compact binaries are unique, unprecedented probes of the strong field and of dynamical aspects of general relativity. More importantly, they give us a brand-new cosmic window to our Universe—gravitational-wave astronomy [1–8].

The capability of searching for GW signals relies on both the sensitivity of GW detectors and the theoretical waveform templates modeled for gravitational-wave sources used in matched-filtering data analysis techniques. Currently, the matched-filtering method is the standard and the most optimal signal processing technique used by the gravitational-wave community to find GW signals from compact binary mergers in noisy detection data. Although weak signal extraction and source information inversion of GW based on matched-filtering techniques are very successful, this has a great weakness as well as a potential hazard. Data analysis through matched filtering has a huge computational cost if the parameter space in question is large. This is the major motivation behind many authors proposing deep learning for GW data analysis [9–12]. Another possible problem with standard matched-filtering techniques is that the completeness and accuracy of the GW waveform template are prerequisites for it to work.

This implies that there is a risk of losing GW signals beyond the theoretical expectation. Yet, identifying GW signals beyond the expectation of the theory will greatly facilitate the development of astronomy and also provide important insight into the problems of fundamental physics such as quantum gravity and physics under extreme conditions [13–15].

Although matched filtering provides an optimal solution for identification of gravitational waves under Gaussian noise, the data in practice from GW detectors contain many non-Gaussian noise transients, also known as “glitches.” A comprehensive classification and characterization of these noise features may provide valuable clues for identifying the source of noise transients and possibly lead to their elimination. Machine learning is becoming more and more important in various disciplines such as particle experimentation [16,17], gamma ray detection [18,19], supernovae classification [20,21], weak lensing data analysis [22–25], source modeling [26–29], and others. There have also been many attempts to use machine learning algorithms in gravitational-wave data analysis to show promise for the noise classification, categorization [30–38], and cancellation [39,40]. Recently, an innovative project called “Gravity Spy”¹ [41,42] has combined the power of machine learning, with the help of volunteers, to label data sets of glitches and create a superior classifier of glitches in LIGO data. Machine learning has been widely used in GW data processing, especially in the identification of signals and the classification of noise.

*Corresponding author;
zjcao@amt.ac.cn

¹See www.gravityspy.org.

In recent years, deep learning, a new area of machine learning research, has been in the spotlight [43]. In the past few years, some researchers have demonstrated empirical success of deep neural networks in the application to data analysis of gravitational waves [44–48]. These researchers include George and his co-workers [9,10], Gabbard *et al.* [49], and others [11,12,50]. These published works used the convolutional neural network (CNN) from different perspectives to identify GW signals with low signal-to-noise ratios (SNR). The works tell us that CNN architecture plays an important role for recognition of CBC GWs (and also continuous GWs [51,52] and CCSNe GWs [53]) in data with simulated or real noisy background from LIGO. Gebhard *et al.* [54] offered an enlightening discussion on the general limitations of CNNs and proposed an alternative CNN-based architecture with proper performance metrics. They also claimed that their trained network can cover all the GW events in both the first and second observation run (O1/O2) of LIGO—except for GW170817, the first observation of GW from a binary neutron star inspiral.

In this paper, we aim to use the deep learning technique to find all of the known GW events in the O1 and O2 data. In addition, we would like to mark other possible GW event candidates. We have trained CNNs like the ones used in [9–12]. Although these neural networks can find GW150914, none of them can find other events in O1. Thus, we adjust the usual CNN a little bit. Based on our adjusted CNN networks, we can find all 11 GW events reported by LIGO [8]. Besides these confirmed events, we have also found 2069 substantial triggers in O1.

The plan of this paper is as follows: In Sec. II, we describe the adjustment of the CNN neural network. Based on our adjusted CNN architecture, training data samples, test data samples, training strategy, and search methodology on the real LIGO recording are described in Sec. III. Subsequently, we apply our trained network to the O1 and O2 data in Sec. IV. Lastly, Sec. V is devoted to a summary.

II. MATCHED-FILTERING CONVOLUTIONAL NEURAL NETWORKS AND SIGNAL RECOGNITION STRATEGY

A. Structure of matched-filtering convolutional neural network

The conventional matched filtering technique uses each waveform in the template bank to perform convolutional matching with the data. Such convolution is frequently called the inner product. The inner product operation is usually done in the frequency domain

$$\langle d|h \rangle = 4 \int_0^\infty \frac{\tilde{d}(f)\tilde{h}^*(f)}{S_n(f)} e^{2\pi i f t_c + \phi_c} df. \quad (1)$$

But it can also be expressed as a convolution in the time domain. Here, d represents data, h represents the theoretical

waveform, and S_n represents the power spectrum of the noise. Since noise always dominates the LIGO data, we use the power spectrum of data d directly to estimate S_n .

Different than in the usual CNN network, we add a sensing layer as the first layer. We further divide the neurons of the sensing layer into N_t groups. The coefficients h_{ij} , $i = 1, \dots, N_t$, of each group are fixed, and they correspond to a whitened theoretical waveform. The index j corresponds to the data sequence of each waveform. After the convolution operation between the input data and each group of neurons (see the details of the operation in the Appendix), we output the maximal value of each convolution. Then, we collect these maximal values as the output of the first layer. The remaining layers are the same as in the conventional CNN networks used in previous work [9,10].

The coefficients h_{ij} are analogous to the template waveforms in matched-filtering data analysis. We use some template waveforms used by LIGO to set these h_{ij} . Our basic idea is to use these specific template waveforms to sense GW signals which are deeply buried in the noise. But unlike the conventional matched-filtering technique, we only need tens of templates here instead of millions of templates. Apparently, if N_t is taken to be the number of waveforms in the template bank, the sensing layer is nothing but the usual matched filtering. However, we are only concerned about an N_t in the tens to hundreds of templates range. Within this range, our tests indicate that the behavior of the network is roughly independent of N_t . In the current work, we use $N_t = 35$, which works quite well in finding signals in the LIGO data.

Logically, our newly designed CNN works in the following way: The coefficients h_{ij} we chose span a subspace of the function space of the GW signal, and after the first layer, the essential matched-filtering operation isolates the signal buried in the noise and projects it into the subspace. Certainly, such a projection will admit some feature structures which may not be recognized by humans but can be recognized by the following CNN layers. This can be used to distinguish the GW signal from pure noise.

Regarding the $N_t = 35$ waveforms used in the first layer of our newly designed CNN network, we choose spinless equal mass binary black holes with total masses $M = 5 + 2i M_\odot$, $i = 0, \dots, N_t - 1$.

Before we feed the data into our CNN model, a Tukey window [55] with $\alpha = 1/8$ is applied to remove edge effects at the beginning and end of the data stretch.

Since our key layer mimics the matched-filtering operation, we call our newly designed CNN network a matched-filtering convolutional neural network (MFCNN).

In the current work, we consider detectors H1 and L1 jointly and input two data streams at the same time into our network. Our matched-filtering CNN network is illustrated in Fig. 1. Here, N represents the length of the input data stream, and L represents the length of each template waveform. Both “template” and “weight” are

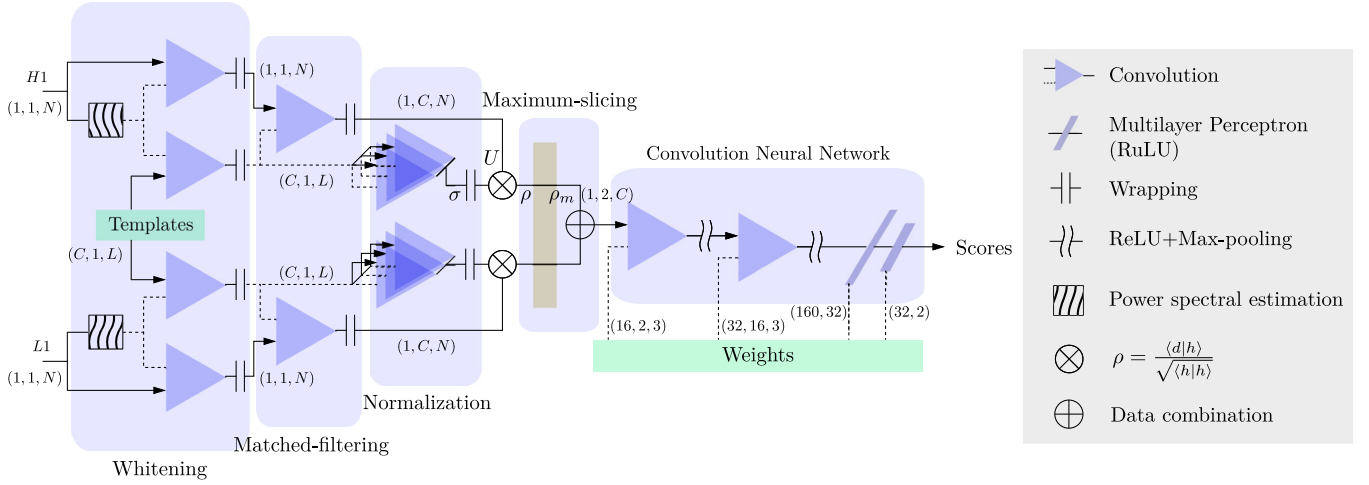


FIG. 1. Structure of the matched-filtering convolutional neural network (MFCNN).

coefficients of the network. But the template coefficients are given by the template waveform, while the weight coefficients are determined through learning.

B. Signal recognition strategy

With a very-long-duration data set D such as LIGO O1 data or O2 data, our task is to identify the data segments including GW signals. After being trained, our MFCNN is ready for such a task. The basic strategy for signal recognition is taking a given period T data segment, say $T = 5$ seconds, as input data and letting the MFCNN process. Then, we move the time window to input different time periods gradually.

In inner product calculations (A2), we need to estimate the power spectrum S_n from the input data d . Since LIGO noise is typically not stationary, it is better to estimate S_n using longer input data. However, longer input data mean a more massive computation requirement. As a compromise, we use 5-second duration data as input, although typically, the GW signal lasts half a second for LIGO. For each run, we take a 5-second-long data segment from D . After each segment, we move forward 1 second to get another 5-second-long data segment. So, each 1-second snippet will appear in five segments and be processed 5 times by our network. If we assume our network can recognize GW signals lasting 1 second in duration, our network will continuously alert 5 times for a true signal lasting 1 second. For a true signal lasting more than 1 second, more than five continuous alerts are expected.

If the output confidence value is bigger than a given threshold value $p > p_c$, our network gives an alert. If more than five continuous alerts happen, a trigger for the GW signal will be given.

Given a trigger, which part of the data should be analyzed further to find that the signal is the practical problem? Based on our algorithm described above, a trigger corresponds to five or more alerts according to the

output confidence value. Since each output alert corresponds to a 5-second data segment, we enlarge the corresponding time of the continuous alerts before the first alert and after the last alert. The data segment corresponding to the enlarged time segment is assumed to contain a GW signal. For record purposes, we output the center time of such a time segment and the duration time.

III. TRAINING AND TEST OF THE NEURAL NETWORK

A. Training data set and test data sets

Each data sample includes a noise part and possibly a signal part. We obtain the O1 data from the Gravitational Wave Open Science Center (GWOSC) [56]. The background noises for training or testing are sampled from O1 data that exclude GW150914, GW151012, and GW151226 events. Together with a simulated GW signal, we construct 3220 samples for training data and test data, respectively. Of these 3220 samples, half are pure noise and the other half include signals.

We use our SEOBNRE model [57] to generate simulated gravitational waveforms. In the current work, we only consider circular, spinless binary black holes. Corresponding to LIGO detectors, we adopt the configuration from LALSimulation² that sets all binary sources at a right ascension of 1 h 23 m 45 s, a declination of 45 degrees, and a polarization angle of 30 degrees and consider the total mass of the two black holes falling in the range $10 M_\odot$ – $150 M_\odot$ with an interval $2 M_\odot$ and a mass ratio $q = m_1/m_2$ ranging from 1–10 with an interval 0.1. Regarding the orbital plane direction, we set $\iota = 0$. We determine the source luminosity distance D based on a given SNR.

²See lscsoft.docs.ligo.org/lalsuite/lalsimulation/group_lalsim_detector_strain.html.

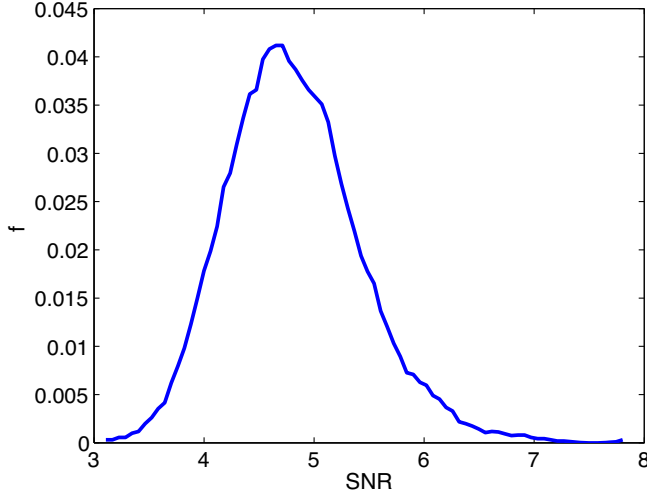


FIG. 2. SNR distribution of the training data set and testing data set for the simulated signals.

We create training and testing data sets, both containing 1610 waveforms. After a transfer training process, the final SNR distribution for the training data is shown in Fig. 2, which has the same distribution of the testing data set. Each data sample consists of two time series that are 5 seconds long with a sampling rate of 4096 Hz. The two time series correspond to detectors H1 and L1, respectively. For the samples with a GW signal, we set the peak location of the signal at the center of the time series. The mass distribution of templates, training or test waveforms, and also the 11 GW events in O1/O2 are plotted in Fig. 3.

B. Training strategy

The coefficients of our adjusted CNN network, except the first layer, are determined through the training process. First, we use the “Xavier” initialization [58] to assign initial random values to these yet-to-be-determined CNN parameters. This initializer is designed to keep the scale of gradients roughly the same in all layers. Then, we use the

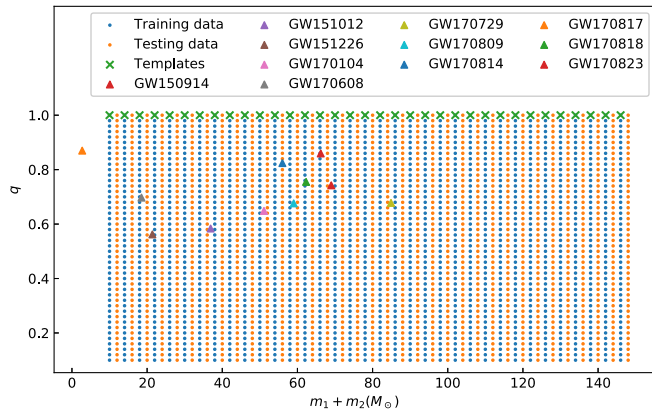


FIG. 3. The total mass and mass ratio of training or test data and templates. The 11 GW events for both O1 and O2 are also shown.

binary output scores s from our network to calculate the confidence for a GW signal using the sigmoid function:

$$p = \frac{1}{1 + e^{-s}}. \quad (2)$$

Next, we use a binary cross-entropy loss function to evaluate deviation between the predicted values and the actual values in the training data. Based on this estimation, a minibatch Adam algorithm [59] is applied to optimize the kernel entries in the CNN. Here, we caution that this confidence value cannot be interpreted as the statistical significance of a detection [54].

Within every training epoch (i.e., a full pass over all training data), not only is the entire training or test data set randomly shuffled but also the background noise is newly resampled in a random manner from O1 data, excluding the three GW events. At the end of every epoch, the performance of the network during training is evaluated based on average accuracy for the networks in each minibatch. We set the learning rate to 0.003 and batch size to 16. During the curriculum learning, we gradually decrease the signal strength, the ratio of the amplitude of the signal to the standard deviation of the noise, of both training data and test data from 1 to 0.02. The training process is accomplished within 6 hours on four NVIDIA GeForce GTX 1080Ti GPUs, each with 11 GB of memory. After training, it takes less than 5 days to process all the O1 data. All the implementations of the current work were coded with PYTHON, based on the MXNet framework [60].

C. Accuracy and efficiency test of the neural network based on simulated data

The authors of [49] have compared the GW signal recognition accuracy and efficiency of CNN networks and conventional matched-filtering techniques. Based on the test data set described in the above subsection, we can calculate the corresponding true alarm probability and false alarm probability.

For the usual problems faced by machine learning, test data include a number of individual samples. As we described in Sec. II B, our test data correspond to longer duration data (5 seconds). The labeled data indicate which time segments include the signal (denoting all such segment sets as D_I) and which time segments are just pure noise (denoting all such segment sets as D_N). After the data processing procedure, the time segments that possibly contain the signal (denoting all such segment sets as D_P) are determined. Accordingly, we can define the true alarm probability and false alarm probability as follows:

$$\text{TAP} \equiv \frac{D_I \cap D_P}{D_I}, \quad (3)$$

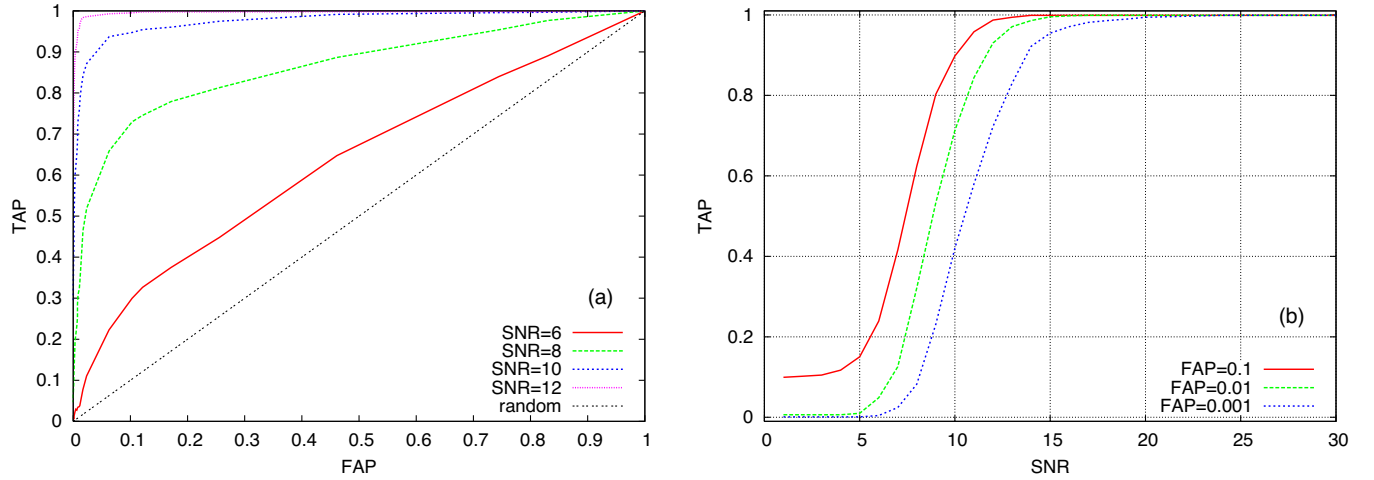


FIG. 4. (a) The ROC curves for test data sets containing signals with matched-filtering SNR 6, 8, 10, and 12, respectively. We plot the true alarm probability versus the false alarm probability estimated by our matched-filtering CNN network. (b) Efficiency curves for false alarm probabilities 0.1, 0.01, and 0.001, respectively. The true alarm probability is plotted as a function of the optimal SNR. In this figure, we use $p_c = 0.5$, $N_t = 35$ and $T = 5$ seconds.

$$\text{FAP} \equiv \frac{D_{\mathcal{N}} \cap D_{\mathcal{P}}}{D_{\mathcal{N}}}. \quad (4)$$

Apparently, $D_{\mathcal{P}}$ will be affected by the choice of p_c . On changing p_c , different TAP and FAP will also be obtained. Accordingly, we can construct the receiver operator characteristic (ROC) curves.

We take some of the O1 data which do not contain any of the three events as pure noise. Then, we randomly inject simulated signals into this pure noise and use the resulting data as test data. When we inject the simulated signal, we adjust the amplitude according to the matched-filtering SNR.

Stronger signals are more easily recognized. Thus, we can see in Fig. 4(a) that larger SNR test data result in better ROC curves. For a given SNR, the true alarm probability can be used to describe the recognition efficiency. For given representative false alarm probabilities 0.1, 0.01, and 0.001, we plot the efficiency curves in Fig. 4(b).

It is interesting to ask how the number of templates N_t and the length of the input data segment T affect the behavior of the MFCNN. Regarding the N_t problem, we compare cases including $N_t = 3, 16, 35, 70, 350$ while evenly locating the signal in the total mass range $[5, 75] M_{\odot}$. We plot the results in Fig. 5(a). We find that even though very few templates, such as $N_t = 3$, are used, the MFCNN can still work well. If N_t is larger than about 20, the behavior is roughly independent of template

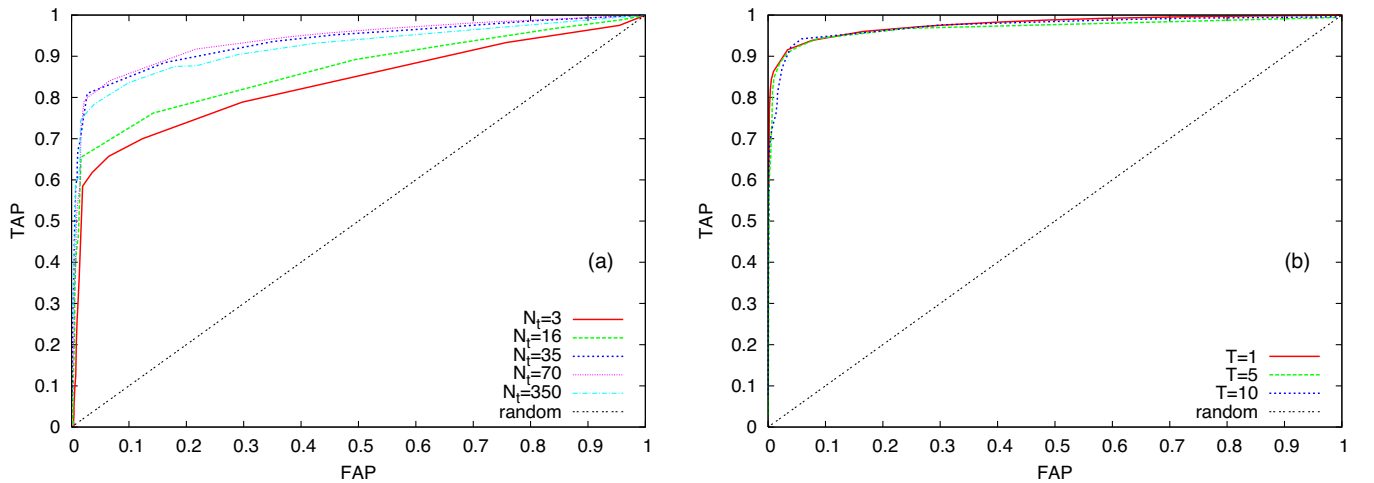


FIG. 5. Comparison of different superparameters of the MFCNN network. (a) Comparison for different numbers N_t of templates. Note that SNR = 10 and $T = 5$ seconds are used. (b) Comparison for different data segment duration T . Here, SNR = 10 and $N_t = 35$ are used.

numbers. So, we fix $N_t = 35$ in the remainder of the current paper.

Regarding T , we compare $T = 1, 5, 10$ seconds, letting the time window advance at intervals of 0.2, 1, 2 seconds, respectively, to ensure five consecutive passes for a signal included in the T data segment. We plot the results in Fig. 5 (b). We can see that the behavior of the MFCNN is roughly independent of the superparameter T . Therefore, we fix $T = 5$ in the remainder of the current paper.

IV. SEARCH RESULTS OF THE REAL LIGO DATA

A. Signal search results of O1/O2

In this section, we apply our MFCNN to LIGO O1 and O2 data. We use $T = 5$ seconds and $N_t = 35$. Regarding the threshold, we use $p_c = 0.5$, which corresponds to TAP ≈ 0.7922 and FAP ≈ 0.0228 for the case SNR = 10 in Fig. 4(a).

If the output confidence value is bigger than a given threshold value $p > p_c$, our network gives an alert. If we

get more than five continuous alerts, a trigger for the GW signal will be given.

In Fig. 6, we plot the confidence values output by our adjusted CNN near the GW150914, GW151012, and GW151226 events for O1. Our network can show all three events clearly. For GW150914 and GW151226, there are five continuous confidence values approaching 1, while other confidence values are much smaller. For GW151012, the confidence values marking the signal are not as big as GW150914 and GW151226, but they are still bigger than 0.5 and also clearly bigger than nearby values.

It is interesting to check the effect of the noise used in the training data set and the real data. To do this, some authors [54] used a network trained with O1 noise to treat O2 data. We do the same test here. We apply the trained network described in the above section directly to O2 data. We show the confidence values for the eight GW events for O2 in Fig. 7. All eight events have been clearly marked. The authors of [54] show that the network trained with a binary black hole coalescence signal may fail to identify the binary neutron star inspiral signal. This is not

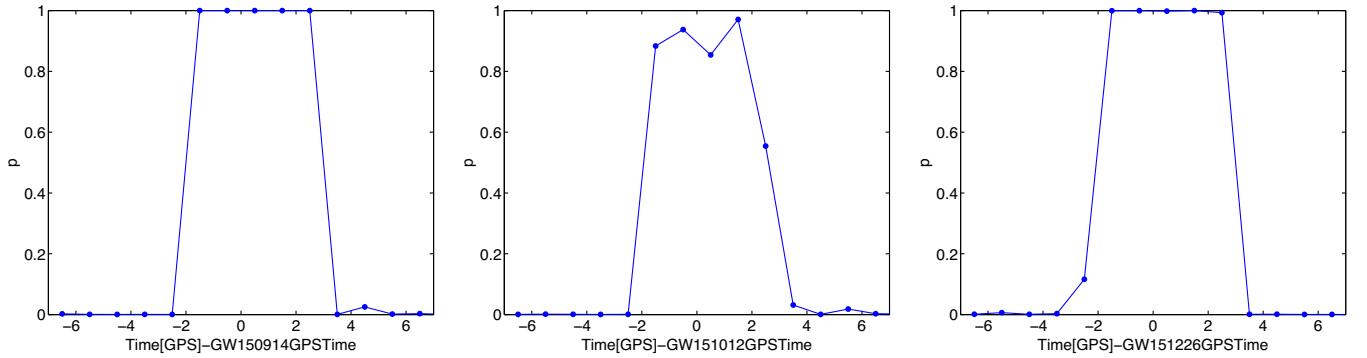


FIG. 6. The output confidence values of our MFCNN near the three GW events of O1.

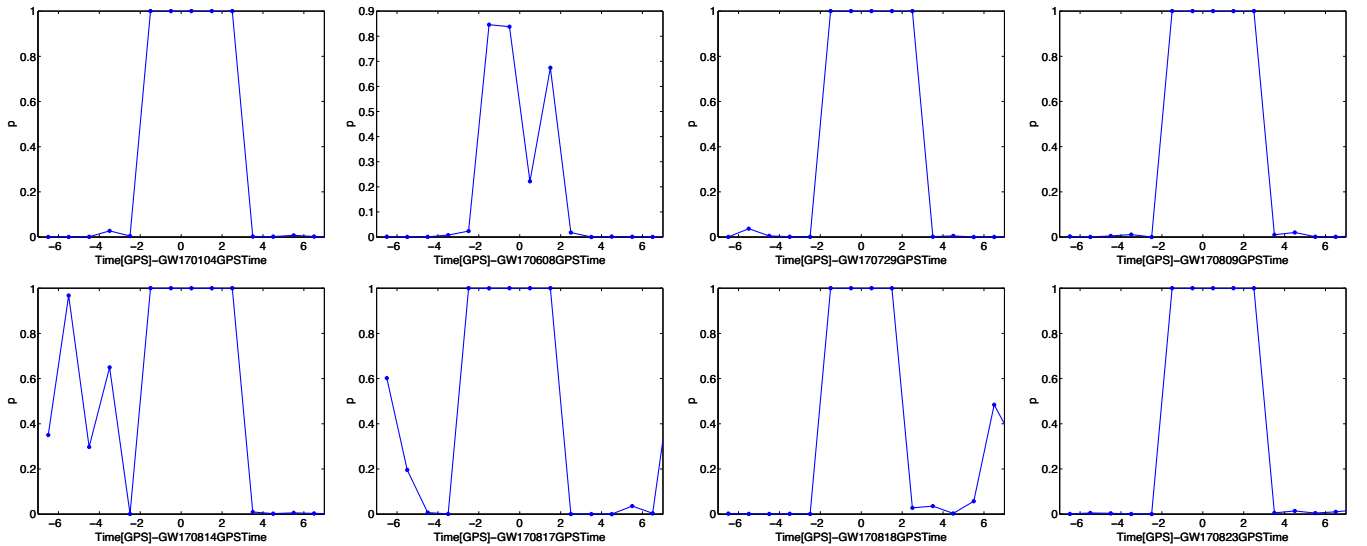


FIG. 7. The output confidence values of our MFCNN trained with O1 noise near the eight GW events of O2.

TABLE I. Numbers and percentage of glitch types involved in O1 data. The distinguishable percentage of the MFCNN is listed in the third row.

	Blip	Koi fish	Chirp	Repeating blips	Others
Numbers in O1	506	247	22	21	16
% in O1	62%	30%	3%	3%	2%
% distinguishable	72%	65%	63%	92%	>99%

true of our adjusted network. We can mark the GW170817 event very clearly.

Besides the three confirmed GW events, we have also applied our network to all of the O1 data. In the current work, we set $p_c = 0.5$. If there are more than five continuous $p > p_c$, we output a signal alarm at the time of the first $p > p_c$.

Ignoring the data quality, we find 3363 triggers in O1, which include GW150914, GW151012, and GW151226. The LIGO data are marked with different tags. The data marked with CBC-CAT3 are not affected by elevated glitches due to environment factors [61]. If we consider only the data marked with CBC-CAT3, there are 2069 triggers in O1. On average, there is one trigger roughly every 40 minutes. As noted by a previous researcher [54], deep learning cannot assign a significance to each trigger. So, we cannot tell which triggers are more believable than others. We can only suggest that these triggers deserve to be checked in more detail with other analysis methods. We have listed these triggered data with center time and time duration on GitHub.³

The authors of GWTC-1 reported three marginal GW event candidates, besides the three known GW events in O1, in Table II of their paper [8]. All three marginal candidates are different from our triggers.

Other authors [62] reported the first open gravitational-wave catalog (1-OGC). Table I of their research [62] listed the 20 event candidates sorted by false alarm rate, which included the three known GW events. All 17 subthreshold candidates differ from our triggers.

We have also checked the consistency between our triggers and the gamma ray burst (GRB) events listed in [63]. There were 1209 GRB events recorded during the O1 run of LIGO. But there were no consistent events found between our triggers and these GRB records.

A recent paper [64] claimed that three more events are found in O2—GW170121, GW170304, and GW170727. Interestingly, we can also find these three events as shown in Fig. 8.

When we increase the threshold value p_c , the number of triggers decreases. For a large $p_c \approx 1$, we get 20 triggers, 15 of which fall in the known glitch catalog of O1 data [65].

One of the remaining five triggers corresponds to GW150914. Compared to the 2069 triggers of $p_c = 0.5$, these 20 triggers admit a much smaller false alarm probability, although the true alarm probability is also dramatically low. The low true alarm probability results in the loss of many true signals in O1. But because of the relatively small false alarm probability, the four remaining triggers deserve a detailed analysis.

These four triggers happened on September 19, 2015, December 19, 2015, December 22, 2015, and January 12, 2016, respectively. We have checked the Rayleigh-statistic spectrum for these. The Rayleigh-statistic spectrums of September 19, 2015, December 19, 2015, December 22, 2015 indicate that the detectors did not work well during those times. On the contrary, the Rayleigh-statistic spectrum of January 12, 2016 indicates the corresponding data were of quite good quality.

Based on a Q-transform analysis, we plot the time-frequency representation of the data strain of the trigger on January 12, 2016 in Fig. 9. The 1-OGC catalog reported 146 214 triggers in all. Among these 146 214 triggers, the one that happened on January 12, 2016 at GPS time 1136593848.9 admits a chirp signal starting from about 1 136 593 846.8 and ending at about 1 136 593 847.7. So, we conclude that our trigger is different from the one reported by 1-OGC. At the same time, we note that 1-OGC reported that the possible binary masses for the January 12, 2016 trigger are, respectively, $m_1 = 4.34 M_\odot$ and $m_2 = 1.65 M_\odot$. From Fig. 9, we can infer that the total mass of the binary is bigger than $50 M_\odot$.

B. Statistical property of the triggers found by MFCNN

When matched filtering is used to extract the gravitational-wave signal, the false alarm rate is an important quantity to indicate its confidence level. Roughly, this false alarm rate is a kind of statistical property of pure noise. It is interesting to compare this property for pure noise to a similar statistical property of the signal candidates. For example, the lines in Fig. 4 of [1] represent the statistical property of pure noise, and the points represent the statistical property of signal candidates.

Although we cannot provide a false alarm rate as given by matched filtering for the MFCNN, we can compare the statistical properties between pure noise and the triggers. We plot such a comparison in Fig. 10. We can see a significant difference between the distribution of the triggers and the noisy O1 data, which implies that the triggers admit different properties than those of the full O1 data set.

C. Distinguishing glitches from the gravitational-wave signal

Known glitches involved in the O1 data have been cataloged in [65]. Glitches often affect the data analysis significantly. So, it is very interesting to explore how our

³See https://github.com/WuShichao/mfcnn_catalog.

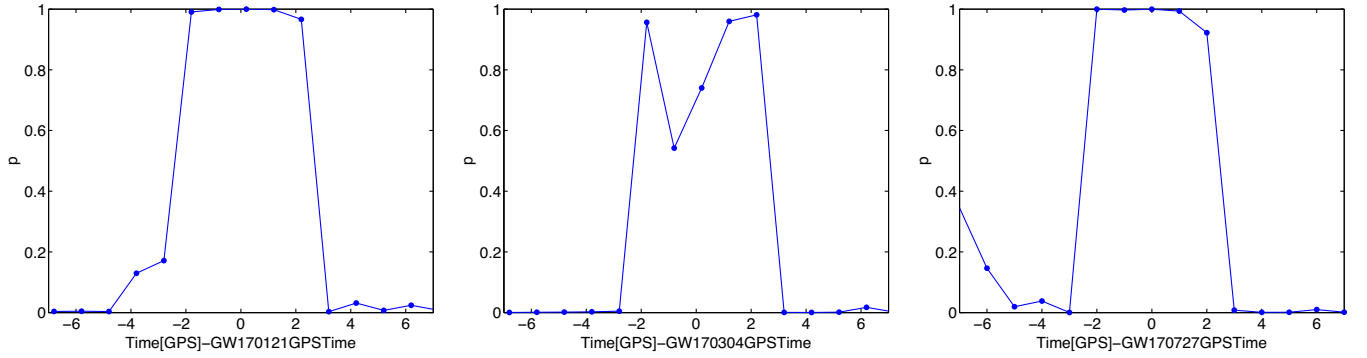


FIG. 8. The output confidence values of our MFCNN near the three GW events of O2 newly reported by [64].

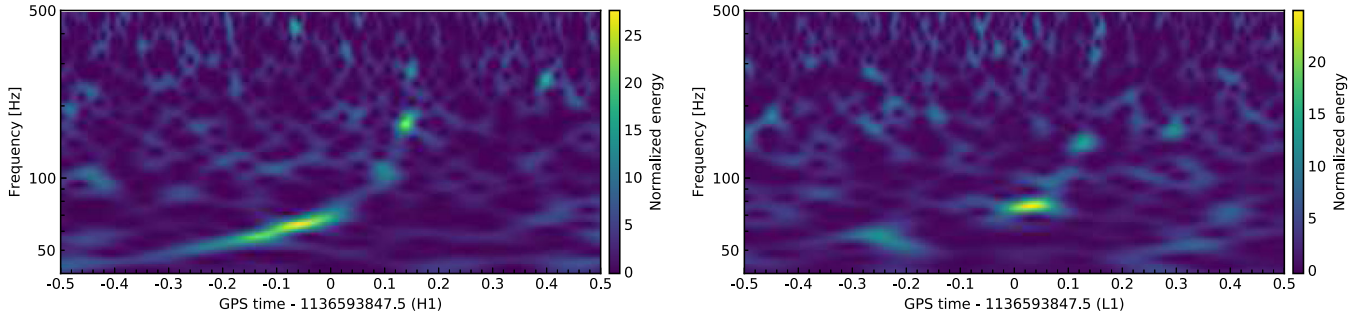


FIG. 9. The spectrograms based on the Q-transform of the two detectors for the trigger on 2016 – 01 – 12. The left diagram is for the Hanford detector, marked with H1. The right diagram is for the Livingston detector, marked with L1.

MFCNN reacts to glitches. When we use our MFCNN to treat the O1 data, we do not consider the glitches at all. If our MFCNN treated some glitches as noise, the network could distinguish the glitch in question from the GW signal. Otherwise, it may confuse the glitch with the GW signal. So, the glitches that fall in our trigger set are those that our

MFCNN cannot distinguish. According to [65], there are 7368 glitches included in O1 data, 812 of which fall in our trigger set or about 10% of known instrumental glitches. Roughly speaking, we find that our MFCNN distinguishes glitches with higher frequency, longer duration, and higher SNR more easily.

According to [65], the glitches in O1 can be catalogued as blip type, koi fish type, and others. The ability of our MFCNN to distinguish different types of glitches differs. We list the numbers and percentage of each type included in O1 in Table I. We also list the percentage accuracy at which our MFCNN can distinguish each type in the third row.

V. SUMMARY

There have been many works published about applying deep learning techniques to gravitational-wave data analysis in the past few years. Most of these works used simulated data. In the current paper, we designed an adjusted CNN and applied it to the entire O1 data of LIGO.

Trained with noise taken from O1 data and simulated binary black hole coalescence waveform, our network can clearly identify the three confirmed GW events. In addition, we used this trained network to directly identify the eight GW events found in O2, and we can also mark all eight events clearly. Although this test is not consistent due to different noise behavior between O1 and O2, our test

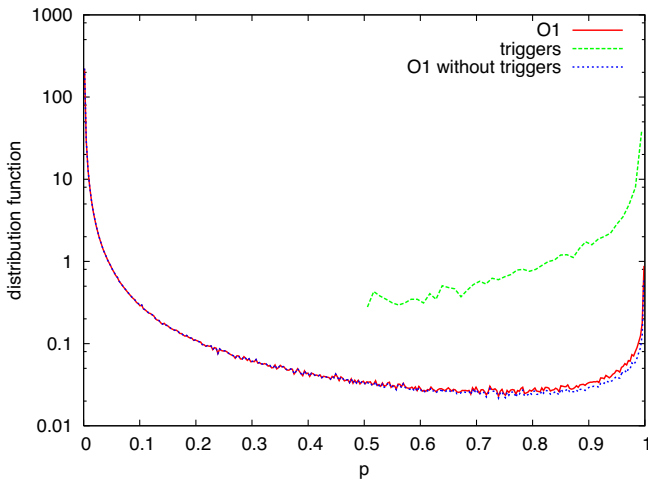


FIG. 10. Comparison of distribution with respect to the MFCNN output p value for triggers, all O1 data, and O1 data without triggers. Note we have chosen $p_c = 0.5$, so the distribution for triggers starts from 0.5.

results indicate that our network model and method are robust with respect to the training data set. A similar investigation has been done in [54].

Besides the three GW events included in O1, we have also found more than 2000 candidates. As noted by the authors of [54], we cannot assign a significance to each candidate. So, we simply call these candidates triggers. These triggered data segments deserve to be investigated in detail.

We have compared our more than 2000 triggers with the subthreshold events reported in GWTC-1 [8] and in 1-OGC [62]. No consistent events were found between these subthreshold events and our triggers. We have also checked the reported GRB events listed in [63] that occurred during the O1 observation time of LIGO. We have not found consistent events between these GRB events and our triggers.

ACKNOWLEDGMENTS

We are thankful to Hao Wei and Jing Li for many helpful discussions. This work was supported by the NSFC (Grants No. 11690023 and No. 11622546) and by the collaborative research program of the Institute for Cosmic Ray Research (ICRR), University of Tokyo. Z. C. was supported by the Fundamental Research Funds for the Central Universities, the Interdisciplinary Research Funds of Beijing Normal University and the Strategic Priority Research Program of the Chinese Academy of Sciences, Grant No. XDB23040100.

APPENDIX: CONVOLUTION OPERATION IN THE FIRST LAYER OF THE ADJUSTED CNN NETWORK

For a template $h(t)$ and detected strain data $d(t)$, the matched-filter SNR is defined as

$$\rho^2(t_c) \equiv \frac{|\langle d|h \rangle(t_c)|^2}{\langle h|h \rangle}, \quad (\text{A1})$$

$$\langle d|h \rangle(t_c) = 4 \int_0^\infty \frac{\tilde{d}(f)\tilde{h}^*(f)}{S_n(f)} e^{2\pi i f t_c} df, \quad (\text{A2})$$

$$\langle h|h \rangle = 4 \int_0^\infty \frac{\tilde{h}(f)\tilde{h}^*(f)}{S_n(f)} df, \quad (\text{A3})$$

where $|\cdot|$ means the absolute value, $*$ is the complex conjugate, and \sim is the Fourier transformation,

$$\tilde{d}(f) = \int_{-\infty}^\infty d(t) e^{-2\pi i f t} dt. \quad (\text{A4})$$

The quantity $S_n(f)$ is the one-sided average power spectral density of the detector noise. Due to the convolution theorem and the relationship between convolution and correlation, the correlation in Eq. (A2) can be rewritten as

$$\begin{aligned} \langle d|h \rangle(t) &= 4 \int_0^\infty \frac{\tilde{d}(f) \cdot \tilde{h}^*(f)}{S_n(f)} e^{2\pi i f t} df \\ &= 4 \int_0^\infty [\tilde{d}(f) \bar{S}_n(f)] \cdot [\tilde{h}(f) \bar{S}_n(f)]^* e^{2\pi i f t} df \\ &= 4 \int_0^\infty \tilde{d}(f) \cdot \tilde{h}^*(f) e^{2\pi i f t} df \\ &= 2\bar{d}(t) * \bar{h}(-t), \end{aligned} \quad (\text{A5})$$

$$\bar{d}(t) = d(t) * \bar{S}_n(t), \quad (\text{A6})$$

$$\bar{h}(t) = h(t) * \bar{S}_n(t), \quad (\text{A7})$$

$$\bar{S}_n(t) = \int_{-\infty}^{+\infty} S_n^{-1/2}(f) e^{2\pi i f t} df, \quad (\text{A8})$$

where $d(t) * h(t)$ means the convolution of functions $d(t)$ and $h(t)$. Similarly, $\langle h|h \rangle$ can also be calculated in this way as $\langle h|h \rangle = 2[\bar{h}(t) * \bar{h}(-t)]|_{t=0}$.

[1] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. Lett.* **116**, 061102 (2016).
[2] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. Lett.* **116**, 241103 (2016).
[3] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. X* **6**, 041015 (2016).
[4] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. Lett.* **118**, 221101 (2017).
[5] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. Lett.* **119**, 141101 (2017).

[6] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. Lett.* **119**, 161101 (2017).
[7] B. P. Abbott *et al.*, *Astrophys. J. Lett.* **851**, L35 (2017).
[8] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. X* **9**, 031040 (2019).
[9] D. George and E. A. Huerta, *Phys. Rev. D* **97**, 044039 (2018).
[10] D. George and E. Huerta, *Phys. Lett. B* **778**, 64 (2018).
[11] X. Li, W. Yu, and X. Fan, *arXiv:1712.00356*.

- [12] X.-L. Fan, J. Li, X. Li, Y. Zhong, and J. Cao, *Sci. China Phys. Mech. Astron.* **62**, 969512 (2019).
- [13] X. He and Z. Cao, *Int. J. Mod. Phys. D* **24**, 1550081 (2015).
- [14] B. Sun, Z. Cao, and X. He, *Sci. China Phys. Mech. Astron.* **62**, 40421 (2019).
- [15] X. He, J. Jing, and Z. Cao, *Sci. China Phys. Mech. Astron.* **62**, 110422 (2019).
- [16] T. Cohen, M. Freytsis, and B. Ostdiek, *J. High Energy Phys.* **02** (2018) 034.
- [17] S. Chang, T. Cohen, and B. Ostdiek, *Phys. Rev. D* **97**, 056009 (2018).
- [18] S. Caron, G. A. Gomez-Vargas, L. Hendriks, and R. R. de Austri, *J. Cosmol. Astropart. Phys.* **05** (2018) 058.
- [19] S. Abraham, N. Mukund, A. Vibhute, V. Sharma, S. Iyyani, D. Bhattacharya, A. R. Rao, S. Vadawale, and V. Bhalerao, *arXiv:1906.09670*.
- [20] T. Charnock and A. Moss, *Astrophys. J. Lett.* **837**, L28 (2017).
- [21] A. Moss, *arXiv:1810.06441*.
- [22] A. Gupta, J. M. Z. Matilla, D. Hsu, and Z. Haiman, *Phys. Rev. D* **97**, 103515 (2018).
- [23] M. Shirasaki, N. Yoshida, and S. Ikeda, *Phys. Rev. D* **100**, 043527 (2019).
- [24] D. Ribli, B. r. Pataki, J. M. Z. Matilla, D. Hsu, Z. Haiman, and I. Csabai, *Mon. Not. R. Astron. Soc.* **490**, 1843 (2019).
- [25] J. Fluri, T. Kacprzak, A. Lucchi, A. Refregier, A. Amara, T. Hofmann, and A. Schneider, *Phys. Rev. D* **100**, 063514 (2019).
- [26] A. J. K. Chua, C. R. Galley, and M. Vallisneri, *Phys. Rev. Lett.* **122**, 211101 (2019).
- [27] A. Rebei, E. A. Huerta, S. Wang, S. Habib, R. Haas, D. Johnson, and D. George, *Phys. Rev. D* **100**, 044025 (2019).
- [28] A. J. K. Chua and M. Vallisneri, *Phys. Rev. Lett.* **124**, 041102 (2020).
- [29] Y. Setyawati, M. Prer, and F. Ohme, *Classical Quantum Gravity* **37**, 075012 (2020).
- [30] S. Rampone, V. Pierro, L. Troiano, and I. M. Pinto, *Int. J. Mod. Phys. C* **24**, 1350084 (2013).
- [31] S. Mukherjee, R. Obaid, and B. Matkarimov, *J. Phys. Conf. Ser.* **243**, 012006 (2010).
- [32] J. Powell, D. Trifir, E. Cuoco, I. S. Heng, and M. Cavagli, *Classical Quantum Gravity* **32**, 215012 (2015).
- [33] J. Powell, A. Torres-Forn, R. Lynch, D. Trifir, E. Cuoco, M. Cavagli, I. S. Heng, and J. A. Font, *Classical Quantum Gravity* **34**, 034002 (2017).
- [34] D. George, H. Shen, and E. A. Huerta, *Phys. Rev. D* **97**, 101501 (2018).
- [35] N. Mukund, S. Abraham, S. Kandhasamy, S. Mitra, and N. S. Philip, *Phys. Rev. D* **95**, 104059 (2017).
- [36] M. Razzano and E. Cuoco, *Classical Quantum Gravity* **35**, 095016 (2018).
- [37] 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).
- [38] D. George, H. Shen, and E. A. Huerta, *Phys. Rev. D* **97**, 101501(R) (2018).
- [39] W. Wei and E. A. Huerta, *Phys. Lett. B* **800**, 135081 (2020).
- [40] H. Shen, D. George, E. A. Huerta, and Z. Zhao, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, New York, 2019).
- [41] 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).
- [42] S. Bahaadini, V. Noroozi, N. Rohani, S. Coughlin, M. Zevin, J. Smith, V. Kalogera, and A. Katsaggelos, *Information Sciences (NY)* **444**, 172 (2018).
- [43] G. Allen *et al.*, *arXiv:1902.00522*.
- [44] H. Gabbard, C. Messenger, I. S. Heng, F. Tonolini, and R. Murray-Smith, *arXiv:1909.06296*.
- [45] 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).
- [46] C. Chatterjee, L. Wen, K. Vinsen, M. Kovalam, and A. Datta, *Phys. Rev. D* **100**, 103025 (2019).
- [47] P. G. Krastev, *Phys. Lett. B* **803**, 135330 (2020).
- [48] H. Shen, E. A. Huerta, Z. Zhao, E. Jennings, and H. Sharma, *arXiv:1903.01998*.
- [49] H. Gabbard, M. Williams, F. Hayes, and C. Messenger, *Phys. Rev. Lett.* **120**, 141103 (2018).
- [50] L.-L. Wang, J. Li, N. Yang, and X. Li, *New J. Phys.* **21**, 043005 (2019).
- [51] C. Dreissigacker, R. Sharma, C. Messenger, R. Zhao, and R. Prix, *Phys. Rev. D* **100**, 044009 (2019).
- [52] F. Morawski, M. Beijer, and P. Ciecielag, *arXiv:1907.06917*.
- [53] P. Astone, P. Cerdá-Durán, I. Di Palma, M. Drago, F. Muciaccia, C. Palomba, and F. Ricci, *Phys. Rev. D* **98**, 122002 (2018).
- [54] T. D. Gebhard, N. Kilbertus, I. Harry, and B. Schölkopf, *Phys. Rev. D* **100**, 063015 (2019).
- [55] F. J. Harris, *Proc. IEEE* **66**, 51 (1978).
- [56] M. Vallisneri, J. Kanner, R. Williams, A. Weinstein, and B. Stephens, *J. Phys. Conf. Ser.* **610**, 012021 (2015).
- [57] Z. Cao and W.-B. Han, *Phys. Rev. D* **96**, 044028 (2017).
- [58] X. Glorot and Y. Bengio, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (MIT Press, Cambridge, 2010), pp. 249–256.
- [59] D. P. Kingma and J. Ba, *arXiv:1412.6980*.
- [60] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, *Astrophys. J.* **891**, 123 (2020).
- [61] J. Slutsky *et al.*, *Classical Quantum Gravity* **27**, 165023 (2010).
- [62] A. H. Nitz, C. Capano, A. B. Nielsen, S. Reyes, R. White, D. A. Brown, and B. Krishnan, *Astrophys. J.* **872**, 195 (2019).
- [63] The Gamma-ray Coordinates Network (GCN), TAN: Transient Astronomy Network, <https://gcn.gsfc.nasa.gov>.
- [64] A. H. Nitz, T. Dent, G. S. Davies, S. Kumar, C. D. Capano, I. Harry, S. Mozzon, L. Nuttall, A. Lundgren, and M. Tpai, *arXiv:1910.05331*.
- [65] S. Coughlin, Updated Gravity Spy Data Set (Version v1.1.0), <https://doi.org/10.5281/zenodo.1476551> (2018).