Artificial intelligence model for gravitational wave search based on the waveform envelope

Cunliang Ma,[1] Wei Wang,[1] He Wang,[2] and Zhoujian Cao[3,4,5,*]

[1]*School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China*
[2]*CAS Key Laboratory of Theoretical Physics, Institute of Theoretical Physics, Chinese Academy of Sciences, Beijing 100190, China*
[3]*Institute for Frontiers in Astronomy and Astrophysics, Beijing Normal University, Beijing 102206, China*
[4]*Department of Astronomy, Beijing Normal University, Beijing 100875, China*
[5]*School of Fundamental Physics and Mathematical Sciences, Hangzhou Institute for Advanced Study, UCAS, Hangzhou 310024, China*

(Received 29 December 2022; accepted 6 March 2023; published 27 March 2023)

In recent years, artificial intelligence technology for gravitational wave data analysis has developed rapidly. In this paper, we put forward a new artificial intelligence model for gravitational wave search. The framework of such a model includes a detection stage and a testing stage. We first use the deep learning technology to extract the envelope information of the gravitational wave candidate and use the coalescence time obtained from the envelope to further confirm the detection results. Within the detection stage, we use wavelet denoising and a special training strategy to improve the performance of the gravitational wave detection model. The lowest false alarm rate of the detection model is about 1.7 per month without the testing stage. When the testing stage is added, the lowest false alarm rate decreases to 0.046 per year. The efficiency of our model is demonstrated by the data obtained from the first, second, and third observing runs of the LIGO-VIRGO-KAGRA collaboration. The search results of confident events on the three observing runs indicate that the efficiency of our deep learning algorithm can achieve 80% of the traditional algorithm based on matched filtering.

DOI: 10.1103/PhysRevD.107.063029

I. INTRODUCTION

In 1916, Albert Einstein predicted the existence of gravitational waves (GWs) based on the theory of general reality [1,2]. Due to the weakness of GW signals arriving on the Earth, there was no effective method to directly observe GW events at that time. On September 14, 2015, the advanced Laser Interferometer Gravitational Wave Observatory (aLIGO) directly detected a GW event, GW150914, for the first time in human history [3]. Recently, aLIGO and advanced Virgo detectors have reported over 90 GW events through their three Observing runs (O1, O2, and O3) [4–6]. These ground-breaking discoveries add GWs to the growing set of detectable cosmic messengers and have opened the era of GW astronomy [7].

Unlike the traditional astronomy, GW astronomy observes the Universe through gravitational waves rather than electromagnetic waves. Compared with electromagnetic wave, a gravitational wave is not easily scattered, dispersed, and absorbed in the process of propagation [8]. If the gravitational wave signal is used to locate the source first, then it can be used to guide the subsequent observation of the electromagnetic counterpart, thus it improves the efficiency of electromagnetic wave detection. Researchers can also cross verify the results of the above joint detection and obtain more astrophysical information to realize multimessenger astronomy (such as the detection of GW170817 [9,10]).

Since the GW signal is buried in strong noise, the signal search is a challenging task. Matched filtering (MF) is the optimal method for the task under the assumption of stationary Gaussian noise. However, the MF method is computationally expensive and time intensive. The pioneering works in 2018 show that the GW detection based on the deep learning (DL) method is more computationally efficient than MF [11,12]. The DL method has achieved great success in the field such as object detection [13–15], natural language processing [16–18], and remote sensing [19–21]. Also, many works investigated the DL method for the GW search [22–28]. Ensemble complex models, such as WaveNet, training with millions of data can achieve no misclassifications for one-month detection [24]. The coincident GW search of two detectors is the effective method for eliminating

*Corresponding author.
zjcao@amt.ac.cn

the false triggers [27,28]. The MLGWSC-1 [29] and Kaggle challenges [30] accelerate the application of machine learning in GW search.

The MF-based and DL-based GW search methods each have their own strengths and weaknesses. The results of the MF method have a high degree of confidence and can reflect some source parameters such as the arrival time of the GW from binary black hole (BBH) coalescence, but the calculation time increases when the number of templates increases [31]. The DL method has a real-time capability, but it is difficult to accurately estimate the coalescence time. The information of the coalescence time can help the subsequent GW data processing such as Bayesian inference [32] and it is also important to determine the sky location of the GW source. If the DL method can be used to achieve high-accuracy coalescence time prediction, then the DL method becomes more powerful for GW signal searching.

The envelope of the pure GW signal has the information of the coalescence time. In this work, we propose an envelope extraction network that takes the whitened strain as input and the envelope of the GW signal as output. The reason for using the envelope instead of the Fourier domain is that the Fourier domain of a signal obtained via Fourier transform or short-time Fourier transform can only exhibit its amplitude and phase as a function of frequency over a certain time period, not in an instant. Furthermore, neither the Fourier transform nor the short-time Fourier transform are capable of accurately reflecting the instantaneous changes of amplitude-frequency and phase-frequency of a signal. The results show that, in most cases, the statistical error of the coalescence time estimated by the network is less than 5 ms, which is much smaller than the sliding window method used in [28]. The idea of the envelope extraction network is motivated by the DL model for GW denoising [33–35]. GW denoising can be achieved by an end-to-end deep learning model, such as WaveNet [34] and Autoencoders [33]. Here we analyze the feasibility of the envelope extraction model. After denoising, the envelope can be got via simple calculation. The envelope value of the GW signal $h(t)$ at time t can be estimated by

$$\sqrt{\hat{h}^2 + (\mathcal{H}(\hat{h}))^2}, \tag{1}$$

where $\hat{h} = \hat{h}(t)$ is the denoised signal and $\mathcal{H}$ denotes the Hilbert transformation

$$\mathcal{H}(x(t)) = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{x(t')}{t - t'} dt', \tag{2}$$

where P means the principal Cauchy value.

Signal searching methods may mistake the background noise as a GW signal. Such mistakes are called false alarms. In the MF-based method for GW signal searching, the subsequent testing process such as a time-frequency test [36] can greatly alleviate the false alarm problem. The DL-based searching method cannot provide a template that the time-frequency test process required. Two works have recently used the strategy of comparing the results of two separated DL detection models for Hanford and Livingston detectors to further test the signal searching result [27,28]. So far, to the best of our knowledge, there is no means to quickly and effectively test the search results of one model. We in the current paper propose a new strategy that utilizes the information of coalescence time to test the result of one model.

We design a GW signal search algorithm based on the DL method together with the aforementioned strategy. The algorithm consists of a GW detection network and two envelope extraction networks, respectively, for Hanford and Livingston detectors. The detection model produces a list of GW triggers. Each GW trigger is tested by the envelope extraction model. We estimate the coalescence time of the GW at each detector. When the difference of the coalescence time is less than 50 ms, the DL algorithm reports a GW trigger.

Almost all of the existing DL models for GW signal searching use whitened data as the input. Differently, we apply a wavelet denoising process after whitening. The results show that under the same true alarm probability (TAP = 0.91), the false alarm probability (FAP) of the detection model using wavelet packet denoising decreases about 97%.

The rest of the paper is organized as follows. In Sec. II, we introduce the process of building data for training and testing. After that, we describe our models in Sec. III. Then we apply our models to the O1, O2, and O3 data of LIGO-VIRGO-KAGRA collaboration in Sec. IV. The last section is devoted to a summary.

II. DATA FOR TRAINING AND TESTING

Two datasets are generated by the open-source project ggwd[1] [22]. One contains only the background noise and the other includes background noise and BBH coalescence signals. A time-domain strain $s(t)$ detected by a single interferometer can be formulated as follows:

$$s(t) = h(t) + n(t), \tag{3}$$

where $n(t)$ is the background noise. Such noise is taken from the O1 data collected by Hanford and Livingston interferometers excluding the identified GW events. $h(t)$ is the BBH waveform generated by the effective one-body numerical relativity waveform model [37]. All of the BBH waveforms contain three phases of BBH coalescence: inspiral, merger, and ringdown.

For each BBH signal, the distance between the Earth and the source is determined by a given network SNR [38], which is randomly sampled in (8, 20). The mass of the two black holes of BBH is randomly sampled in $(5M_\odot, 80M_\odot)$

[1]https://github.com/timothygebhard/ggwd.

and the dimensionless spin is randomly sampled in (0, 0.998). The polarization angle and the coalescence phase are sampled uniformly at random from the interval $(0, 2\pi)$. The inclination angle is sampled uniformly at random from the interval $(0, \pi)$. The right ascension and the declination are sampled from a uniform distribution over a sphere. The duration of each strain is 16 s and the sampling rate is 4096 Hz. The coalescence time of every strain locates at 8 s. Each strain is whitened and treated through a high-pass filter with a cutoff frequency of 20 Hz.

Two data sets are constructed, which obey the same parameter distribution. The only difference is that the background noise of the two datasets comes from the Hanford and Livingston interferometers, respectively. Every dataset includes 256,000 samples, of which 96,000 contain GW signals and 160,000 contain pure background noise. This dataset is divided into three parts: the original training set includes 160,000 samples (80,000 GW and 80,000 noise), the original validation set includes 16,000 samples (8000 GW and 8000 noise), and the original test set includes 80,000 samples (8,000 GW and 72,000 noise). Note that the ratio of GW samples to pure noise samples in the test set is not 1 to 1, but 1 to 9. The purpose of this division is to mimic the fact that the number of gravitational wave samples is much lower than the number of noise samples in the real detection process.

III. AI MODEL FOR GW SIGNAL RECOGNITION

In this section, the proposed artificial intelligence (AI) model for GW signal recognition will be systematically introduced. The workflow of MF-based GW detect pipelines [39–41] can be roughly divided into two stages: obtaining the triggers from interferometers and then verifying them. For example in PyCBC, triggers are generated by thresholding and clustering the SNR time series obtained from interferometers then all the triggers need to pass the coincidence and data quality tests. Motivated by the MF based pipelines, we also adopt the multistage scheme for the AI model.

The overall framework of the AI model as shown in Fig. 1 is briefly divided into three stages: GW detection, envelope extraction, and coalescence time testing. The raw data is whitened and then denoised by the wavelet packet (WP) denoising in the preprocessing stage. In the WP denoising, the high frequency components of the whitened strain is eliminated and only the low frequency components will be used as input to the detection model. The detail description about the WP denoising is given in Sec. III A 1. The detection model analyzes the strains from Hanford and Livingston simultaneously. The output of the detection stage reflects the confidence of a GW signal candidate. These continuous outputs form the preliminary detection results. The detection model will issue an alarm when its output exceeds a threshold and a GW trigger will be given if five or more consecutive alarms are generated. When there

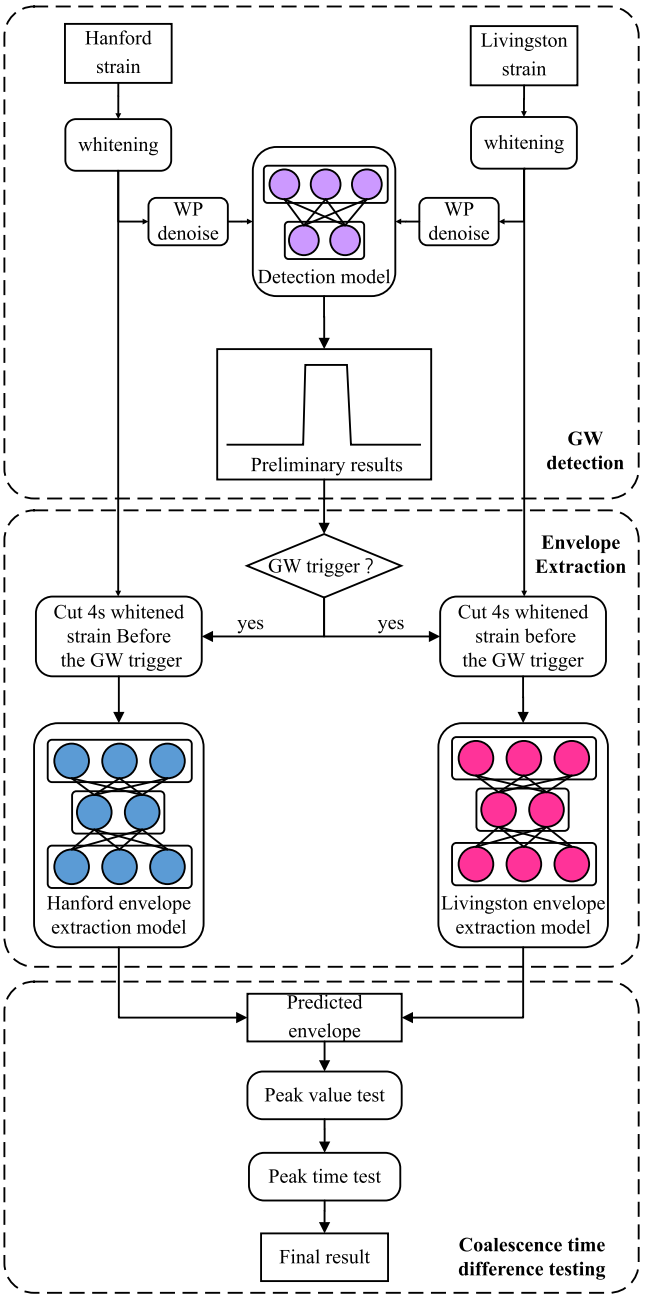

FIG. 1. Schematic diagram of the proposed AI model for a gravitational wave search based on waveform envelope. The black dotted box represents the three main stages of the model.

is a GW trigger in the preliminary detection result, the two envelope extraction models analyze the 4 s whitened strain from the Hanford and Livingston interferometers, respectively, and output the normalized envelope predictions corresponding to the GW signal. The input window size of the detection model is 1 s while the input window size of the envelope extraction model is 4 s. They are different, and the end of the two input windows are aligned. After obtaining the envelope predictions from Hanford and Livingston, the GW trigger needs to pass the coalescence

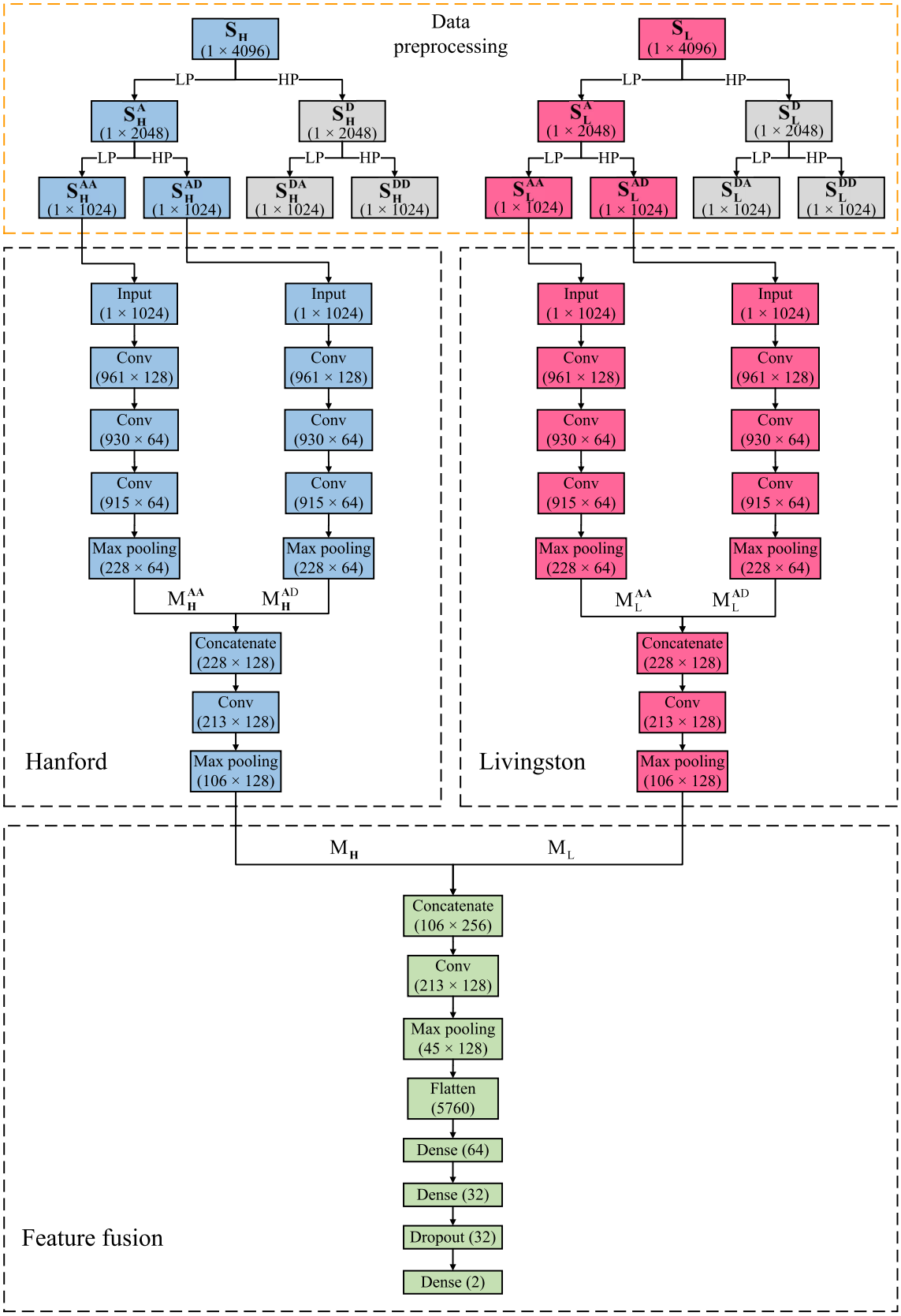

FIG. 2. Schematic diagram of the detection stage in our AI model (Fig. 1). LP and HP represent the low-pass filter and high-pass filter. A and D represent the approximation and detail components of the strain, respectively. In the rectangular boxes, "Conv" represents the convolutional layer and the numbers in the boxes represent the shape of the matrix output by this layer.

time difference testing. Only the GW triggers that pass the coalescence time difference test will be retained as the final detection result.

A. Detection stage

The schematic diagram of the detection stage is shown in Fig. 2. The whole detection stage is divided into two parts: data preprocessing and DL model detection. In the following, we will introduce data preprocessing, model structure, and training strategy in detail.

1. Data preprocessing

Wavelet packet decomposition (WPD) is a commonly used signal processing method and is widely used in the fields of audio signal processing [42–44] and image recognition [45–47]. In the field of GW data analysis, some researchers use the WPD to convert one-dimensional strain into multi-dimensional matrixes [48,49]. In our work, the whitening strain is decomposed into two levels by successively translating and convolving the Daubechies wavelet with low-pass (LP) and high-pass (HP) filters. LP and HP filters retain the approximation (A) and detail (D) components that contain low frequency and high frequency information of the strain. Suppose the whitened strain of interferometer X (H for Hanford and L for Livingston) is denoted as $\mathbf{S}_X \in \mathbb{R}^{4096}$ (1 s data with 4096 Hz sampling rate). After the process of WPD, $\mathbf{S}_X$ is decomposed to $\mathbf{S}_X^P \in \mathbb{R}^{1\times 1024}$ ($P \in \{AA, AD, DA, DD\}$). AA, AD, DA, and DD represent four branches of the

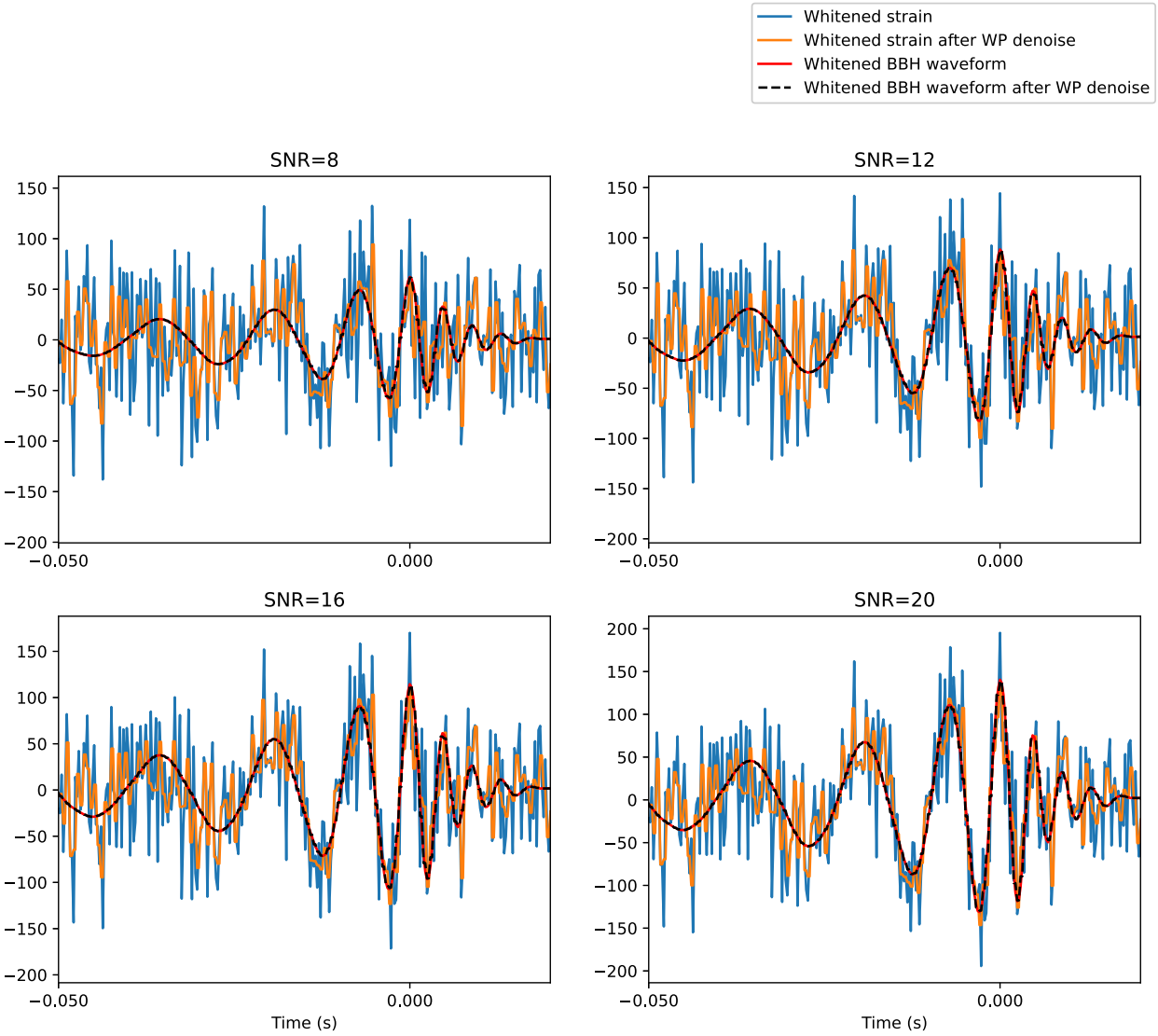

FIG. 3. The effect of wavelet packet denoising on noise and signal, respectively. Here the simulated GW signal is for BBH with parameters $m_1 = 36M_\odot$, $m_2 = 36M_\odot$. The four subplots admit the SNR 8, 12, 16, and 20, respectively. The noise is taken from the whitened background noise of Hanford detector.

decomposition tree. Different from [48,49], we discard the $\mathbf{S}_X^{DA}$ and $\mathbf{S}_X^{DD}$, which contain the high frequency message of $\mathbf{S}_X$. The above strategy can be regarded as a special WP denoise for the whitened strain. The frequency of GW signal produced by the BBH is around hundreds Hertz and the whitened noise admits frequency range from 20 to 2048 Hertz. So we only focus on the low frequency components to sift out part of the noise while keep the GW signal intact. In the next section, we will analyze the effect of the special WP denoising. We will find that the denoising does enhance the signal relative to the noise.

2. The effect of wavelet packet denoising

We do wavelet packet transformation first and then reconstruct the time sequence through an inverse wavelet packet transformation base on the low frequency components S_X^{AA} and S_X^{AD}. To illustrate the effect of WP denoising on noise and signal respectively, we plot the noise and GW waveform before and after WP denoising in Fig. 3. Here both relative weak signal and strong signal are considered. From Fig. 3, we can clearly observe that the background noise is reduced by the WP denoising while the GW signal changes little.

We use the ratio of energy SNR (*RESNR*) before and after the denoising to quantitatively evaluate the effectiveness of the wavelet packet denoising method. The *RESNR* is defined as

$$RESNR = \frac{AESNR}{BESNR}, \tag{4}$$

where *AESNR* and *BESNR* are the energy signal to noise ratio (*ESNR*) after and before the denoising, respectively. The energy signal to noise ratio *ESNR* can be calculated as

$$ESNR = \frac{\int_T |h(t)|^2 dt}{\int_T |n(t)|^2 dt}, \tag{5}$$

where $h(t)$ is the buried signal, $n(t)$ is the background noise, and T is the time interval with 1 s duration. We have set the coalescence time at 0.75 s in our tests. The *RESNR* is higher than 1 means that the denoising preprocess is effective. We construct 20,012 samples to test *RESNR*. The mass of BBH is sampled in the range $(5M_\odot, 80M_\odot)$ and the sampling step is $1M_\odot$. The SNR of samples is in $\{8, 10, 12, 14, 16, 18, 20\}$. Figure 4 shows the distribution of *RESNR* of the samples. All of the *RESNR* values of the experiment are higher than 1.6, with the majority of results falling between 1.8 and 2.1. The data demonstrated in Fig. 4 illustrate that the WP denoising method is effective. The experimental results in Sec. III also show that this WP preprocessing method can significantly improve the performance of the model under low FAP range.

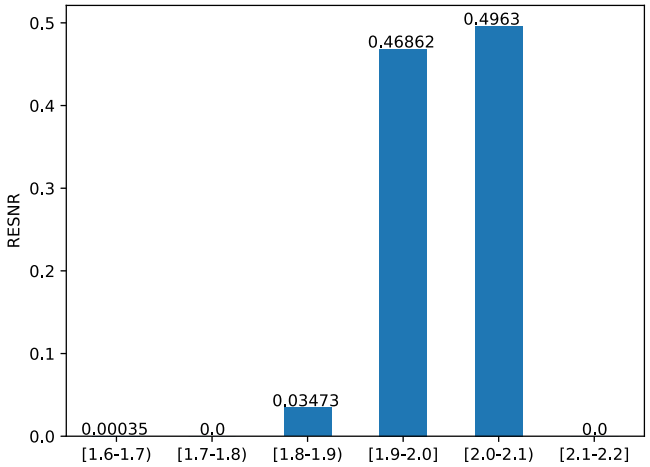

FIG. 4. The ratio of energy SNR *RESNR* distribution for wavelet packet denoising among the 20,012 tested samples.

3. The formulation of the detection stage

Inspired by the design idea of detection model in [24], our detection model contains two submodels that analyze the data from Hanford and Livingston interferometers, respectively. Since the branches of WP decomposition tree include messages of different frequency bands, each branch is analyzed by independent subnetworks.

In the following, we give a brief description of the detection stage. First, we feed the decomposed strain of interferometer X $\mathbf{S}_X^P \in \mathbb{R}^{1\times 1024}$ to a subnetwork N_X^P with three convolutional layers and one max pooling layer. The subnetwork N_X^P can be formulated as

$$\mathbf{M}_X^P = N_X^P(\mathbf{S}_X^P, \mathbf{W}_X^P), \tag{6}$$

where $\mathbf{M}_X^P \in \mathbb{R}^{228\times 64}$ is the output matrix of the subnetwork that contains the features of different frequency bands and $\mathbf{W}_X^P$ represents the subnetwork's trainable weights. $P \in \{AA, AD\}$ represents different branches of the decomposition tree.

Second, the features extracted from different frequency bands are analyzed together. For each interferometer X, all the $\mathbf{M}_X^P$ are concatenated together and passed through a second subnetwork SN_X. The process can be written as

$$\mathbf{M}_X = SN_X(\mathbf{M}_X^{AA} \oplus \mathbf{M}_X^{AD}, \mathbf{W}_{SX}), \tag{7}$$

where $\oplus$ denotes the concatenate, $\mathbf{M}_X \in \mathbb{R}^{106\times 128}$ is the output matrix of the second subnetwork for interferometer X, and $\mathbf{W}_{SX}$ is the trainable weights.

Finally, the features from two interferometers are used to predict whether there is a GW signal. The $\mathbf{M}_H$ and $\mathbf{M}_L$ will be concatenated and processed with a network N_O with one convolutional layer and three fully

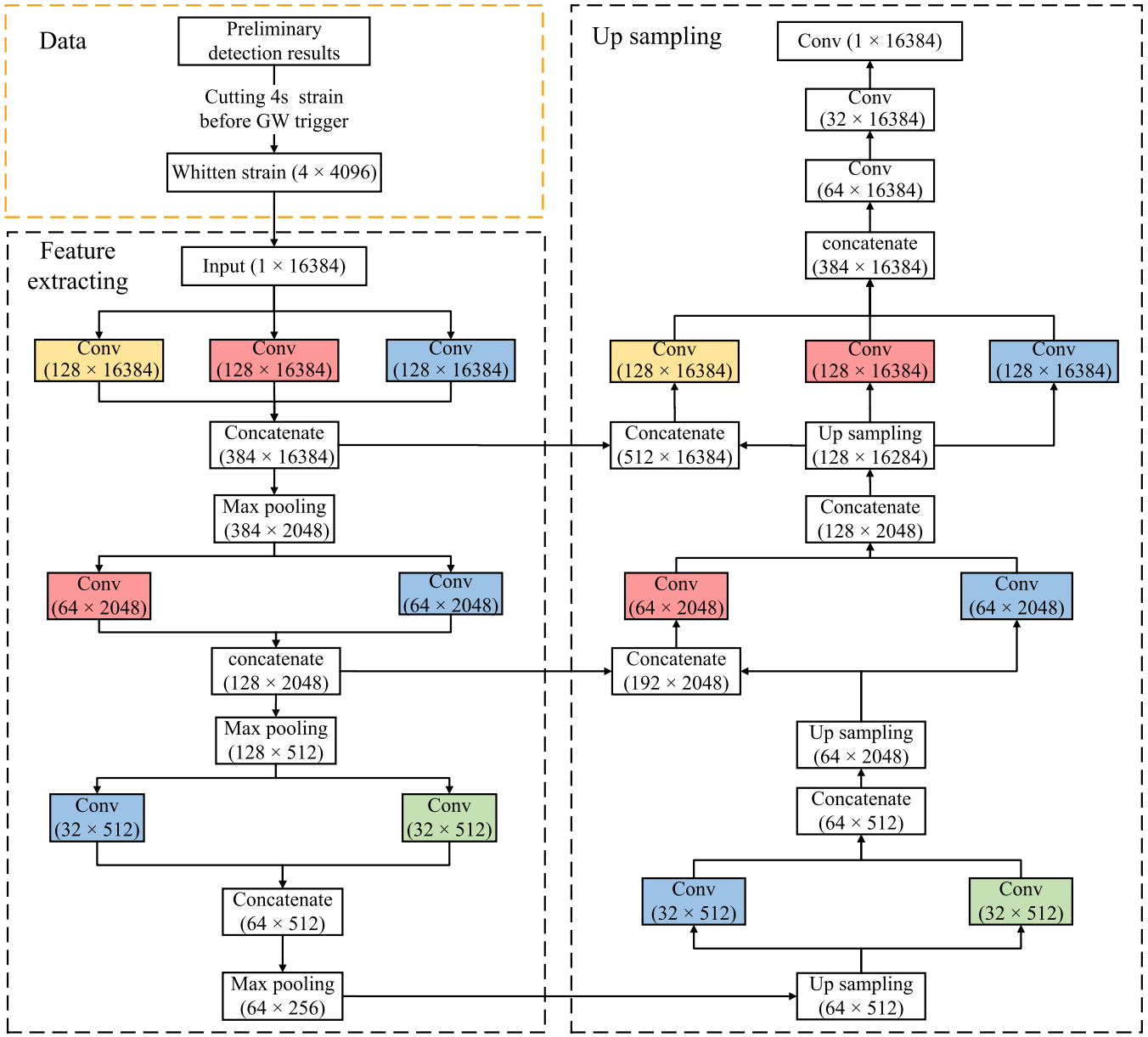

FIG. 5. The network structure of the envelope extraction stage of the AI model in the current work. The solid rectangles represent different layers in the model. The numbers in the rectangles represent the shape of the vector output by the corresponding layer. The yellow, red, blue, and green convolutional layers represent convolutional layers of different receptive fields whose kernel size is 64, 32, 16, and 8, respectively.

connected layers. The output of the detection stage is then given by

$$\mathbf{O} = N_O(\mathbf{M}_H \oplus \mathbf{M}_L, \mathbf{W}_O), \tag{8}$$

where $O \in [0, 1]^2$ is the output of the detection stage. That is to say that the detection stage outputs two time sequences. Each sequence takes value in [0,1].

Before the output layer, a dropout [50] layer is used. The dropout layer can improve the generalization ability of the model and the dropout proportion in this model is 0.5. Except for the last dense layer, the nonlinear activation functions of all hidden layers are Elu. The activation function used in the last dense layer is Softmax, which can be written as

$$\mathrm{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=0}^{n} e^{z_j}}, \tag{9}$$

where n is the number of categories of the classification problem, i is a certain class, the final output of the function is an n-dimensional vector, each element in the vector represents the confidence of a class, and the sum of all elements

is 1. In this paper, n is 2, and the first element p_{score} in the output vector represents the confidence of a GW candidate.

4. The training scheme of the detection network

Compared with the training scheme of the base learner in our previous work [28], the training scheme used in this work has three improvements: training by order of SNR range (inspired by [51]), data balance, and data augmentation. First, the GW strain in the original training set is equally divided into four parts according to the SNR range [(8, 11) (11, 14), (14, 17), (17, 20)], and other parameters follow the same distribution for the four SNR ranges. Different from the Ref. [51], in our work, the SNR is randomly distributed over the corresponding range of each part. The detection model is gradually trained from the samples in the low SNR part to the high SNR part. Second, although the ratio of noise strain to GW strain is 1:1 in the training set, this ratio is random in each training batch. Hence, we set the ratio of noise strain to GW strain to 1:1 in each training batch. Third, we have improved the data augment method compared to our previous work [28]. The main difference between the improved data augment method and the previous one is that the previous one is used before training, while the improved one is used during training. For the improved one, all the strain in each batch is randomly cut to 1 s and ensures that the coalescence time in every GW strain is within $(\frac{1}{8}s, \frac{7}{8}s)$, which means that the coalescence time of GW strain is different in each epoch. Assuming that the training epoch is n, theoretically, the number of strain can be enhanced to n times.

The loss function is binary cross-entropy [52], which is used to evaluate the deviation between the predicted values and the actual values in the training set. The Adam algorithm [53] is applied to optimize the model parameters and the learning rate is about 1.5×10^{-5}.

B. Test stage

1. Envelope extraction network

The ith whitened strain in the training set is $\mathbf{s}_i \in \mathbb{R}^{16384}$ (4 s strain with 4096 Hz sampling rate). Each $\mathbf{s}_i$ contains a simulated GW signal that can be denoted as $\mathbf{h}_i \in \mathbb{R}^{16384}$. Suppose the envelope of each GW signal in the training set is $\mathbf{h}_i^{\text{env}} \in \mathbb{R}^{16384}$. The goal of the extraction model is to find a parametrized function $\mathbf{f}(\cdot|\mathbf{W})$ to estimate the signal envelope, where $\mathbf{W}$ is a trainable weight and can be optimized as

$$\mathbf{W}^* = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{f}(\mathbf{s}_i|\mathbf{w}) - \mathbf{h}_i^{\text{env}}\|^2, \tag{10}$$

the $\mathbf{h}_i^{\text{env}}$ can be calculated via the modulus of the analytical signal of $\mathbf{h}_i$ as

$$\mathbf{h}_i^{\text{env}} = \sqrt{\mathbf{h}_i^2 + (\mathcal{H}(\mathbf{h}_i))^2}, \tag{11}$$

where $\mathcal{H}(\cdot)$ denotes the Hilbert transformation.

Envelope extraction can be considered as a special denoising process of gravitational waves. We borrow the design idea from the Unet [54] and the Inception module [55] to build the envelope extraction network, which is shown in Fig. 5. The whole network can be divided into two parts: feature extracting part and up sampling part. The length of the output strain and the input strain are both 4 s. Since the task of the testing stage is to verify the detection result, only the strain containing a GW signal is used to train the network.

The core idea of inception is to concatenate multiple convolutional layers of different kernel sizes. Such operation increases the width of the network on the one hand. On the other hand, since the receptive fields [55] of different branches are different, multiscale information can be extracted. We use three types of inception modules in the processing of feature extraction and up sampling. The yellow, red, blue, and green convolutional layers in Fig. 5 represent different receptive fields whose kernel size is 64, 32, 16, and 8, respectively. The padding of all convolutional layers is set to the same. Consequently the convolutional layer during the convolution operation will pad the input vector to ensure the size of the output vector of different convolutional layers can be concatenated. During the process of feature contracting, the model extracts the gravitational wave features in the GW strain through three different inception modules. These features will be used as the input of the up sampling stage and guide the model to restore the envelope of GW waveform in the up sampling stage. Except the last convolutional layer, the activation functions of all convolutional layers are Elu. The tanh activation is added after the last convolutional layer. The loss function is mean squared error which is a commonly used loss function in deep learning regression problems. It reflects the average error between the predicted value and the true value, which can reflect the degree of extraction of the GW waveform by the model. During model training, we only keep the model with the smallest validation loss.

2. The training scheme of the envelope extraction network

When training the envelope extraction network, only the GW strain in the original training set is used. The SNR of these strain is the same as the SNR distribution of the data used to train the detection network. The envelope extraction network is also gradually trained from samples in the low SNR range part to the high SNR range part. The envelope extraction model adopts the same data augmentation method as the detection model. In order to achieve the purpose of time-sensitive envelope extraction, the GW strain in the training set is randomly cut into 4 s strain and the coalescence time of the BBH waveform is randomly located in (2 s, 4 s).

Meanwhile, the output label for the 4 s strain is its corresponding normalized BBH waveform. The gradient descent strategy used in this model is Adam, the learning rate is 1×10^{-5}. The code implementation of the above models is based on Keras framework [56]. All of the computations are done on a NVIDIA 3060 GPU.

3. Coalescence time difference testing

Since the time difference of the detected BBH coalescence signals between the ground-based interferometers is very close (several milliseconds), we use this difference to further test the preliminary results of the detection model. Suppose the outputs of the envelope extraction model for Hanford and Livingston are $\hat{h}_H^{\mathrm{env}}(n)$ and $\hat{h}_L^{\mathrm{env}}(n)$, respectively, where $n \in [1, 16384]$). The coalescence time testing includes two steps: peak value test and coalescence time difference test. If the peak value of $\hat{h}_H^{\mathrm{env}}(n)$ or $\hat{h}_L^{\mathrm{env}}(n)$ is lower than 0.5, the corresponding GW triggers will be discarded. After passing the peak value test, the coalescence time of Hanford and Livingston will be predicted via

$$t_X = \triangle t \times \arg\max_n \hat{h}_X^{\mathrm{env}}(n), \tag{12}$$

where $\triangle t$ represents the sampling period of the strain (the time duration between two successive sample points), t_X is the predicted coalescence time for the corresponding interferometer and $X \in \{H, L\}$. If the difference between the t_H and t_L is greater than 50 ms the GW trigger will also be discarded.

IV. AI MODEL PERFORMANCE

To illustrate the advantages of the WP denoise and the training strategy, another two comparative models were constructed. Compare to the AI model proposed in the last section (hereafter we call it model III for convenience), model I does not discard the high frequency part ($\mathbf{S}_X^{DA}$ and $\mathbf{S}_X^{DD}$) of the strain after WPD and contains four more subnetworks (N_H^{DA}, N_H^{DD}, N_L^{DA}, N_L^{DD}). Equation (7) for model I becomes

$$\mathbf{M}_X = SN_X(\mathbf{M}_X^{AA} \oplus \mathbf{M}_X^{AD} \oplus \mathbf{M}_X^{DA} \oplus \mathbf{M}_X^{DD}, \mathbf{W}_{SX}).$$

Model I adopts the same training strategy as model III.

Model II has the same structure as model III. But instead of the training scheme introduced in Sec. III, the training strategy of model II lets the SNR of input strain randomly distributed and does not apply data augmentation method to the coalescence time issue. The ratio of noise strain to GW strain is random in each training batch. The characteristics of the three models are listed in Table I.

In the following, we will compare the performance of models I, II, and III and test the envelope extraction model's prediction effect on the coalescence time of GW. Then, we test our AI model's performance on real GW events reported by LVK in O1, O2. And more we also analyze the corresponding false alarm rate (FAR) of the AI model in the one-month duration detection. Finally, we analyze the detection results of the AI model in O3 events under the threshold of FAR = 2 per year. Since the sampling rate of the datasets in MLGWSC-1 and Kaggle (2048 Hz) differs from the rate used in our model (4096 Hz), our detection model cannot be directly applied to the available datasets of MLGWSC-1 and Kaggle.

TABLE I. The characteristics of three models for the comparison of the WP denoise and the training strategy.

	WP denoise	Training strategy
Model I	⋯	√
Model II	√	⋯
Model III	√	√

A. Result based on the test dataset

1. Performance of the detection network

In the strain data recorded by LIGO, the proportion of background noise is much larger than that of GW signal. Even if the detection model has a relatively low FAP, numerous false triggers may still be generated in the case of long-duration detection. To realize a reliable and efficient DL search pipeline, it is crucial to build a detection model with high TAP in the case of very low FAP. Therefore, we focus on the ROC curves in the low FAP range. We build a test dataset from the original test dataset. The test data set contains 80,000 samples (8000 GW signals and 72,000 pure noise) and the duration of every sample is 1s. The coalescence time of all GW samples is randomly located in ($\frac{1}{8}$ s, $\frac{7}{8}$ s). The ROC curves of models I, II, and III are drawn in the range FAR $\in [0, 0.005]$, which are shown in Fig. 6.

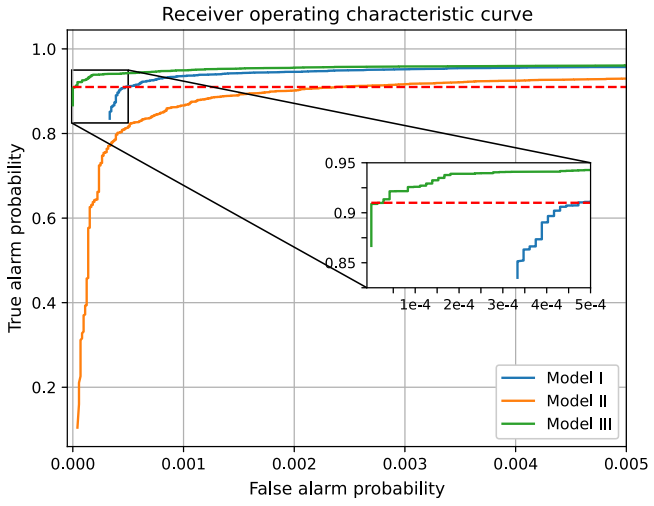

FIG. 6. ROC curves for the test dataset. The TAP of the red dashed line is 0.91. The blue, orange, and green lines represent the ROC curves of models I, II, and III.

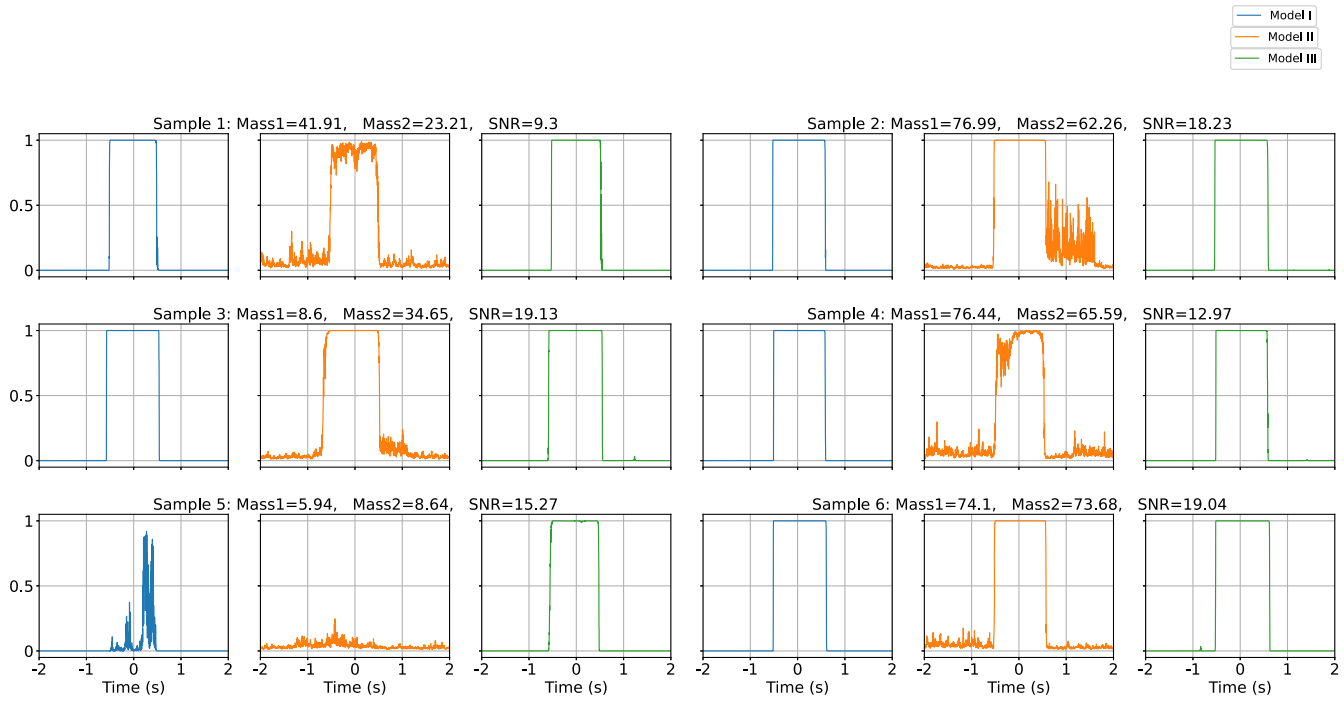

FIG. 7. The output values of models I, II, and III with six samples in testing data. The sliding time step is $\Delta t = 2$ ms.

Comparing models II and III, when the TAP is about 0.91, the FAP of models II and III are about 2.33×10^{-3} and 1.39×10^{-5}, respectively. That is to say that, with the same TAP, the FAP of model III improves more than two orders than that of model II. It is an interesting finding that using the training scheme described in Sec. III can significantly improve the performance of the model at low FAP. Comparing models I and III, when the TAP is about 0.91, the FAP of model II is about 4.72×10^{-4}. This means that with the same TAP, the FAP of model III improves more than one order than that of model I. Note that model I has a more complex structure and more parameters than model III with WP denoising. However, under the same TAP, the FAP of model I is much higher. This fact indicates that using WP denoise can make GW signal identification more reliable than simply using WPD.

Next, we compare the detection performance of the three models on samples randomly selected from the test data set. For real GW detection, detection model needs to process long-duration data. We input a data segment with duration 1 s into the detection model for GW signal recognition, then move forward a time step Δt and use the following data segments for the following works. Figure 7 shows the model's response of six GW samples with 4 s duration. We let the GW event locates in the center of the corresponding strain. After that, models I, II, and III are used to detect these strain, and the Δt is set to 2 ms. The GW signal recognition results of models I, II, and III are shown in Fig. 7 and the physical parameters of the GW sample are shown in the titles of the subplots.

Comparing the GW signal recognition results, we find that the output of model III is stable and close to 1 in the time period when GW exists and close to 0 in the time period when there is only background noise, which means that this model has stronger detection ability than models I and II. Interestingly, the output of models I and III is close to 0 when there is only background noise.

2. *Performance of the envelope extraction network*

In the following, we exemplify the results from the envelope extraction network on the original test set that was introduced in Sec. II. Note that the duration of samples in the original test set is 16 s, so we randomly cut the samples to 4 s and let the coalescence time locate in (3 s, 4 s). Figure 8 shows the output of the envelope extraction network for eight samples randomly selected from the test dataset. We find that the both outputs of Hanford and Livingston can efficiently catch the information of the envelope of GW signal buried in the background noise, especially near the coalescence time. From the samples 1 and 2, we find that the envelope extraction network is less effective for the early inspiral phase, but behaves quite well for the merger phase.

We take the time of the envelope peak as the predicted coalescence time. The envelope extraction network can be used to predict the arrival time of the BBH merger signal to the corresponding interferometers. There is a time difference in the coalescence time between two interferometers, which is in the order of milliseconds. We can use such time differences to test the preliminary result of the detection model. Certainly, only the error of the coalescence time is

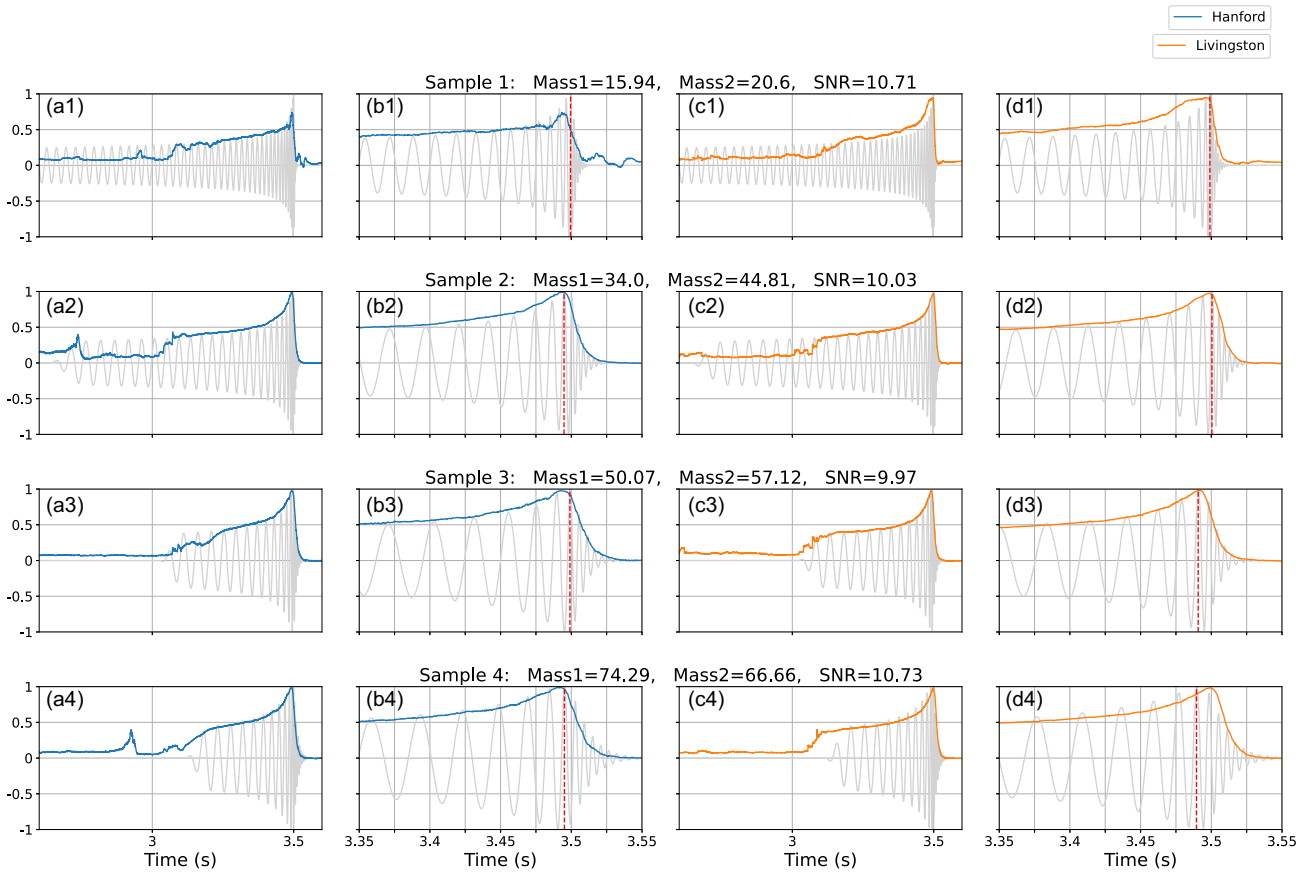

FIG. 8. The output of the envelope extraction network for data strains from Hanford (a), (b) and Livingston (c), (d). The four rows correspond to four samples. Panels (a) and (c) show the whole 1 s time duration. Respectively, corresponding to (a) and (c), (b) and (d) are the enlargement of the merger part including 0.2 s time duration.

small enough, and this information can be applied to the subsequent test for GW detection. We test all the samples in the test dataset and count the coalescence time by using the envelope extraction network. More than 91% and 84% samples on Hanford and Livingston pass the peak value test, respectively. We analyze the samples that pass the value test. The error distribution between predicted coalescence time and true coalescence time is shown in Fig. 9, and we find that more than 92% samples admit an error less than 5 ms. Compared to the coalescence time prediction accuracy in our previous work (about 0.25 s), which used the sliding window method [28], the accuracy of this work (about 5 ms) increased about one order.

B. Result based on the O1 and O2 data

Now we apply our AI model to the O1 and O2 data of Hanford and Livingston interferometers. In the detection stage, we use the time sliding method with a time step $\Delta t = \frac{1}{8}$ s, which means that the detection model detects every $\frac{1}{8}$ s and the detection window is 1 s.

Many researchers use threshold 0.5 [23,26] for signal detection. To compare the performance of models I, II, and III under a same TAP, we set the threshold of the three models by fixing the TAP to 0.91. The thresholds corresponding to the three models are calculated through the ROC shown in Fig. 6. The related thresholds are shown in Table II. When the output value of the detection model

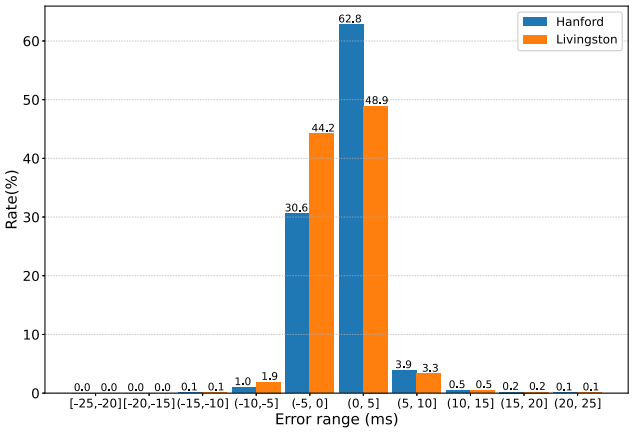

FIG. 9. The distribution histogram of the prediction error of the coalescence time obtained by the envelope extraction network. Blue line and orange line are, respectively, for Hanford and Livingston interferometer. Note that we choose the sample, which passes the peak value test as the total samples.

TABLE II. The resulted thresholds of the three detection models for a GW signal alarm. These thresholds correspond to TAP 0.91.

	Model I	Model II	Model III
Threshold	0.90934	0.71741	0.99995

exceeds its corresponding threshold, an alarm will be given. When the number of continuous alarms exceeds or equals to five, a GW trigger will be generated [23].

1. GW events detection

In the following, we investigate the performance of our AI model on all reported GW events of O1 and O2 by LVK. For each event, we take 16s-long strain and let the GW event locate at the center of the corresponding strain. We compare the performance of models I, II, and III. The detection results output by the three models are shown in Fig. 10.

In Fig. 10, all the BBH events are successfully recognized by models I and III. As we described before, although the detection model is trained only by O1 data, it can work well for O2 data. However, model II fails to recognize GW151226 and GW170818. We find that the outputs of models I and III are close to 0 during the period without GW. This good performance on real data is similar to that on simulated data as shown in Fig. 7. Since the binary neutron star coalescence data do not exist in the training set, none of the three models can recognize GW170817. Although model I generates a GW trigger near GW170817, the duration of the GW trigger matches the glitch near GW170817 [9,10] detected by the Livingston interferometer. Hence, we blame this trigger on the glitch. Comparing

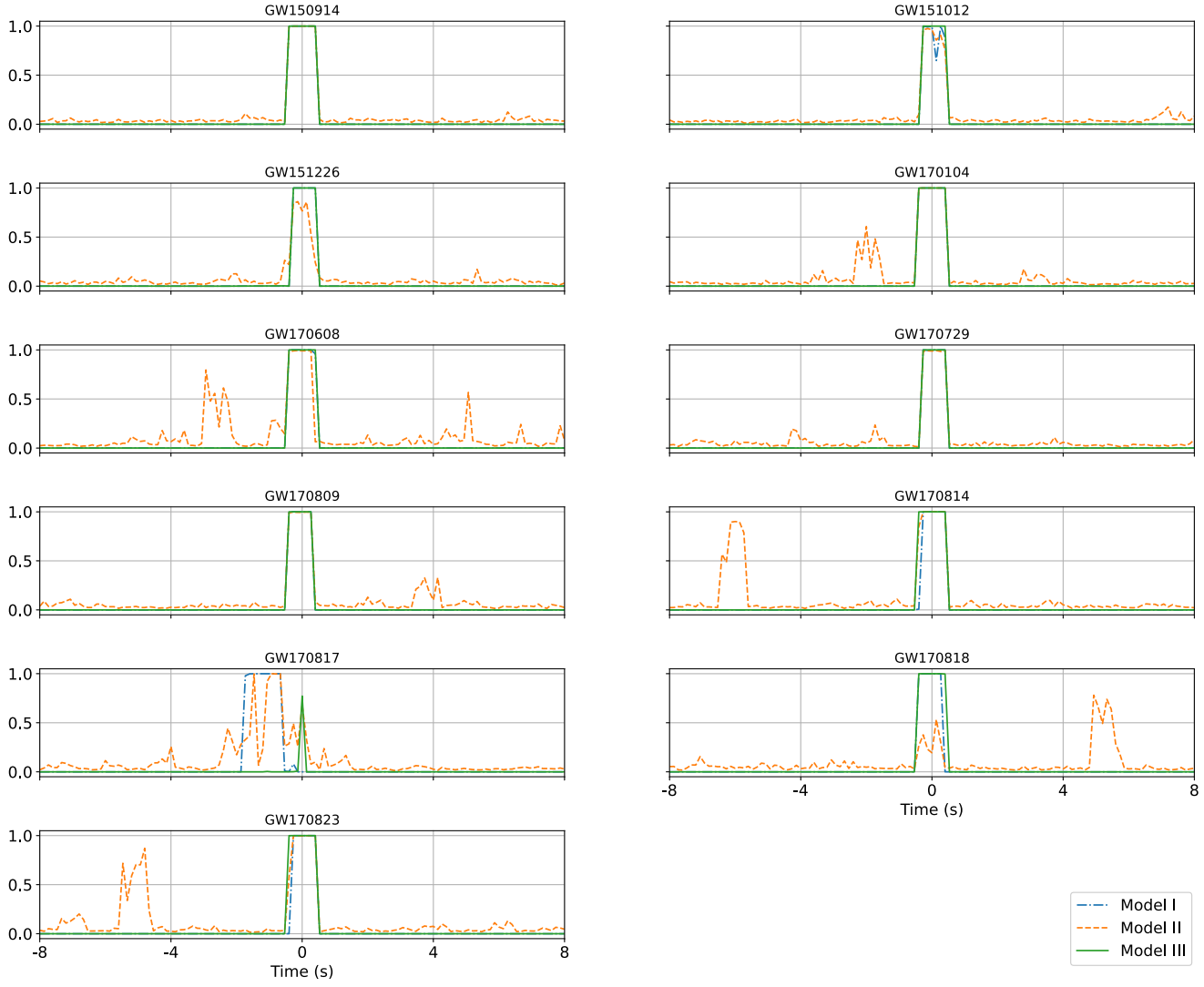

FIG. 10. The GW signal recognition results for the eleven events of O1 and O2 reported by LVK. All GW events happened at 0 s. The results for three models listed in Table I are compared in the plots.

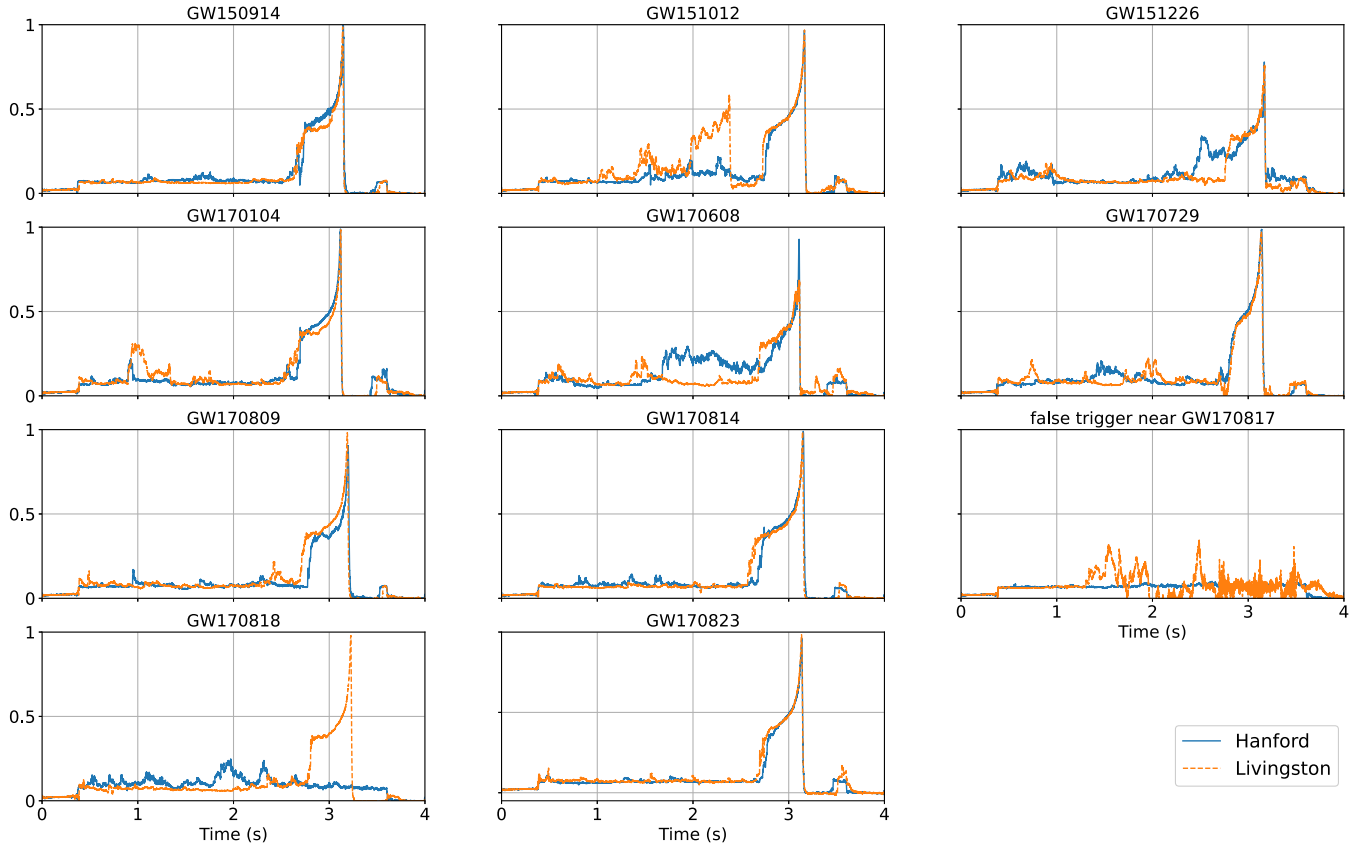

FIG. 11. The envelope extraction results for the eleven events of O1 and O2 reported by LVK. The blue and orange lines correspond to the envelopes for Hanford and Livingston interferometers, respectively.

the GW signal recognition results of models I, II, and III, we find that model III not only successfully recognizes all BBH events in O1 and O2 but also gives out a remarkable subthreshold trigger for GW170817. In addition, model III does not generate any false triggers. Even for the glitch near GW170817, model III can distinguish it from a real GW signal. These results indicate that model III has a better detection capability than models I and II.

The corresponding envelope extraction results for each event are shown in Fig. 11. Except GW170817 and GW170818, the envelope extraction network outputs a very good envelope of the GW signal. Regarding to the strong glitch of GW170817, our envelope extraction network can filter such glitch perfectly. The output envelope of GW170817 is small. We understand this result as that our model is trained only against BBH without a binary neutron star. Regarding to GW170818, the envelope of GW signal on the strain of Hanford interferometer is very small. We blame this result to the weakness of the signal. The matching SNR of GW170818 in Hanford according to PyCBC is just 4.1.

2. Signal search in the whole August 2017 LIGO data

In the evaluation of a detection method performance, the FAR is a significant indicator. FAR reflects the average number of false triggers generated per unit time. Wei *et al.* used two models with different sampling rates to achieve a FAR of about 11.1 per month [25]. Marlin *et al.* used an unbounded soft max replacement to solve the problem of numerical rounding error and reduced the FAR to 1 per month under the test dataset with Gaussian background noise simulated from the aLIGOZeroDetHighPower power spectral density [51]. To investigate the FAR of our AI model, we apply our AI model to one-month duration data on August 2017 of LIGO. The FAR can be calculated by [51]

$$\mathrm{FAR} = \frac{N_f}{T}, \tag{13}$$

where T is the detection duration and N_f is the number of false triggers generated in whole duration. During the one-month detection, models I and II generate 453 and 1898 false triggers. Surprisingly, model III only generates two false triggers. The coworking time of Hanford and Livingston interferometers is about 17.325 days, and the predicted FARs of the three detection models are shown in Table III. The FAR of model III is extremely lower than that of models I and II. Above results indicate that WP denoise and the new training scheme introduced in Sec. III can significantly improve the performance of the detection model.

After testing by the extracted envelope, the false triggers generated by models I, II, and III are, respectively,

TABLE III. The FARs of models I, II, and III in the one-month detection corresponding to the whole August 2017 LIGO data.

	Model I	Model II	Model III
FAR (per month)	784.4	3286.6	3.5

eliminated to 1, 4, and 0. This result indicates the ability of the coalescence time testing scheme. Equation (13) can be used to calculate the FARs of models I and II together with the testing stage but cannot be used to estimate the FAR of model III together with testing stage which has $N_f = 0$. Here we analyze the FAR of model III by Bayesian theory. We use a random variable $x_{FT} \in \{0, 1\}$ to represent whether the detection model generates a false trigger or not, and use another random variable $x_{SE} \in \{0, 1\}$ to represent whether a trigger passes the following envelope test. Assume that the FAR of the detection model in the background noise is FAR_D and the FAR of the total AI model is FAR_{DE}. It is easy to get

$$\mathrm{FAR}_{DE} \propto P(x_{FT} = 1, x_{SE} = 1). \tag{14}$$

Based on the product rule of probability, Eq. (14) can be reduced to

$$\mathrm{FAR}_{DE} \propto P(x_{FT} = 1) \times P(x_{SE} = 1|x_{FT} = 1). \tag{15}$$

Because of

$$\mathrm{FAR}_D \propto P(x_{FT} = 1), \tag{16}$$

we get

$$\mathrm{FAR}_{DE} = \mathrm{FAR}_D \times P(x_{SE} = 1|x_{FT} = 1). \tag{17}$$

Based on the LIGO data of August 2017 our detection network gets two false alarms within about 17.32 days that can be used to calculate the $\mathrm{FAR}_D = \frac{2}{17.32}$. Because the models I, II, and III admit exactly the same envelope network and testing stage operation, we suppose that the probability $P(x_{SE} = 1|x_{FT} = 1)$ is the same for all three models. We can estimate such probability as

$$P(x_{SE} = 1|x_{FT} = 1) = \frac{N_{AFT}}{N_{FT}}, \tag{18}$$

where N_{AFT} represents the number of false triggers after the testing, and N_{FT} represents the number of false triggers before the testing. From model I we have

$$P(x_{SE} = 1|x_{FT} = 1) \approx \frac{1}{453}. \tag{19}$$

From model II we have

$$P(x_{SE} = 1|x_{FT} = 1) \approx \frac{4}{1898} \approx \frac{1}{474}. \tag{20}$$

So we averagely use $\frac{1}{460}$ to estimate $P(x_{SE} = 1|x_{FT} = 1)$. Consequently model III admits a FAR of about 0.046 per year, or equivalently about once 21.83 years.

Above mentioned FAR for model III corresponds to the threshold setting listed in Table II. Since different thresholds correspond to different FARs and different TAPs, we calculate the FAR of model III under different thresholds. Combining these results with $P(x_{SE} = 1|x_{FT} = 1) \approx \frac{1}{460}$ we can get the FAR for the whole AI model of model III. We plot the relation between the FAR and the TAP for model III in Fig. 12. When the threshold of model III is set to

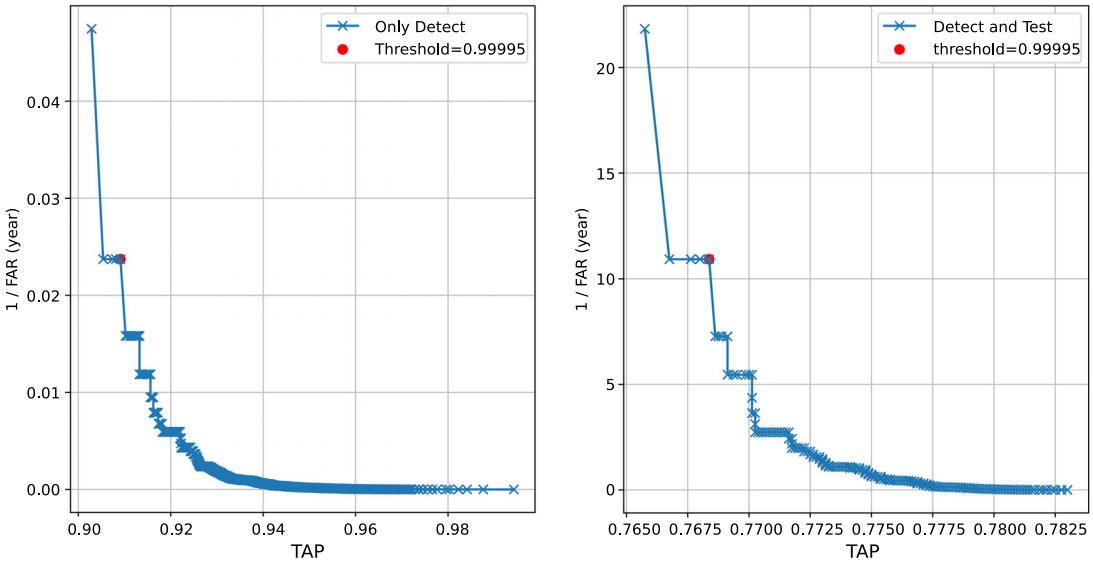

FIG. 12. The relation between FAR and TAP for model III when the p_{score} threshold is changing. Left panel is for detection model only. The right panel is for the whole AI model including both detection stage and test stage.

TABLE IV. The predicted coalescence times for the three events in August 2017 reported by LVK. The listed LVK predicted values correspond to the intermediate value of the posterior distribution given by GWTC-1. The listed OGC predicted values correspond to the intermediate value of the posterior distribution given by 2-OGC. H and L mean Hanford and Livingston. Δ means the time difference between H and L. Based on the posterior distribution given by GWTC-1, the standard deviations for GW170809, GW170814, and GW170823 are, respectively, 0.003279303881529099, 0.0019400082824698507, and 0.017506253479340875.

Event	AI prediction	LVK prediction	OGC prediction
GW170809 (H)	1186302519.7495117	1186302519.7524414	1186302519.756348
GW170809 (L)	1186302519.7382812	1186302519.7436523	1186302519.74707
GW170809 (Δ)	0.0112305	0.0087890625	0.0092778
GW170814 (H)	1186741861.5268555	1186741861.5371094	1186741861.538574
GW170814 (L)	1186741861.519287	1186741861.5292969	1186741861.530273
GW170814 (Δ)	0.007568359375	0.0078125	0.00830101
GW170823 (H)	1187529256.5114746	1187529256.5161133	1187529256.518555
GW170823 (L)	1187529256.5092773	1187529256.5170898	1187529256.519531
GW170823 (Δ)	0.002197265625	−0.0009765625	−0.000976

0.99995, the $\mathrm{FAR}_{DE} \approx 0.0075$ per month, which means that our AI model III generates a false trigger in about 11.41 years. When the threshold of model III is set to near 1, our AI model III can still identify almost all of the BBH events during August 2017 except GW170818. And the corresponding FAR is $\mathrm{FAR}_{DE} \approx 0.0038$ per month, or equivalently 0.046 per year.

The AI, LVK and OGC predictions of the coalescence times for the three BBH events in August 2017 are shown in Table IV. It can be observed from the table that the coalescence times predicted by the AI model for the three events in August 2017 are consistent with the posterior distribution provided by GWTC-1 and 2-OGC.

C. Application of AI model to the O3 data

Next, we investigate the detection performance of our AI model on the O3 data. Model III is chosen as the detection model. There are 44 and 35 confident events reported by LVK in O3a and O3b observing runs, respectively [5,6]. Among the confident events in O3a and O3b, there are, respectively, 38 and 32 events for which the Hanford and Livingston interferometers both working. Tables V and VI

TABLE V. GW events reported by LVK in O3a [5]. Here only events for which both Hanford and Livingston interferometers work well are listed. The notation √ for pipelines cWB, MBTA GstLAL, PyCBC_broad, PyCBC_BBH means the corresponding event passed the FAR threshold or the p_{astro} threshold. The √ for the detection of our AI model means the corresponding event results in detection network output bigger than the threshold of 0.9. The √ for the test of our AI model means that the event passed the coalescence time difference testing. The √ of D&T means the event passed the FAR = 2 per year threshold. The detection rate means the ratio of events number denoted by √ and the total events number listed in the table.

	cWB		MBTA		GstLAL		PyCBC		PyCBC_BBH		AI model		
Name	FAR	p_{astro}	FAR	p_{astro}	FAR	p_{astro}	FAR	p_{astro}	FAR	p_{astro}	Detection	Test	D&T
GW190403_051519	…	…	…	…	…	…	…	…	…	√	…	…	…
GW190408_181802	√	√	√	√	√	√	√	√	√	√	√	√	√
GW190412	√	√	√	√	√	√	√	√	√	√	√	√	√
GW190413_052954	…	…	…	…	…	…	…	…	√	√	…	…	…
GW190413_134308	…	…	√	√	…	…	…	…	√	√	…	√	…
GW190421_213856	√	√	√	√	√	√	…	√	√	√	√	√	√
GW190426_190642	…	…	…	…	…	…	…	…	…	√	…	…	…
GW190503_185404	√	√	√	√	√	√	√	√	√	√	√	√	√
GW190512_180714	√	√	√	√	√	√	√	√	√	√	√	√	√
GW190513_205428	…	…	√	√	√	√	…	…	√	√	√	√	√
GW190514_065416	…	…	…	…	…	…	…	…	…	√	√	√	√
GW190517_055101	√	√	√	√	√	√	√	√	√	√	√	√	√
GW190519_153544	√	√	√	√	√	√	√	√	√	√	√	√	√
GW190521	√	√	√	√	√	√	√	√	√	√	√	√	√
GW190521_074359	√	√	√	√	√	√	√	√	√	√	√	√	√

(Table continued)

TABLE V. *(Continued)*

	cWB		MBTA		GstLAL		PyCBC		PyCBC_BBH		AI model		
Name	FAR	p_{astro}	FAR	p_{astro}	FAR	p_{astro}	FAR	p_{astro}	FAR	p_{astro}	Detection	Test	D&T
GW190527_092055	···	···	···	···	√	√	···	···	···	···	···	···	···
GW190602_175927	√	√	√	√	√	√	√	√	√	√	√	√	√
GW190701_203306	√	√	···	√	√	√	√	√	√	√	√	√	√
GW190706_222641	√	√	√	√	√	√	√	√	√	√	√	√	√
GW190707_ 093326	···	···	√	√	√	√	√	√	√	√	√	√	√
GW190719_215514	···	···	···	···	···	···	···	···	√	√	√	√	√
GW190720_000836	···	···	√	√	√	√	√	√	√	√	···	···	···
GW190725_174728	···	···	···	√	···	···	√	√	√	√	···	···	···
GW190727_060333	√	√	√	√	√	√	√	√	√	√	√	√	√
GW190728_064510	···	···	√	√	√	√	√	√	√	√	√	√	√
GW190731_140936	···	···	···	√	√	√	···	···	√	√	···	√	···
GW190803_022701	···	···	···	√	√	√	···	···	√	√	···	√	···
GW190805_211137	···	···	···	···	···	···	···	···	√	√	√	···	···
GW190814	···	···	√	√	√	√	√	√	···	···	√	√	√
GW190828_063405	√	√	√	√	√	√	√	√	√	√	√	√	√
GW190828_065509	···	···	√	√	√	√	√	√	√	√	···	···	···
GW190915_235702	√	√	√	√	√	√	√	√	√	√	√	√	√
GW190916_200658	···	···	···	√	···	···	···	···	···	√	√	···	···
GW190917_114630	···	···	···	···	√	√	···	···	···	···	···	···	···
GW190924_021846	···	···	√	√	√	√	√	√	√	√	···	···	···
GW190926_050336	···	···	···	···	√	√	···	···	···	···	···	√	···
GW190929_012149	···	···	···	√	√	√	···	···	···	···	···	√	···
GW190930_133541	···	···	√	√	√	√	√	√	√	√	···	···	···
Detection_rate (%)	39.47	39.47	60.53	76.32	76.32	76.32	57.89	60.53	76.32	84.21	60.53	68.42	55.26

TABLE VI. Similar to Table V but for O3b [6]. The notation convention is the same as Table V.

	cWB		MBTA		GstLAL		PyCBC		PyCBC_BBH		AI model		
Name	FAR	p_{astro}	FAR	p_{astro}	FAR	p_{astro}	FAR	p_{astro}	FAR	p_{astro}	Detection	Test	D&T
GW191103_012549	···	···	···	···	···	···	···	√	√	√	···	···	···
GW191105_143521	···	···	√	√	···	···	√	√	√	√	√	···	···
GW191109_010717	√	√	√	√	√	√	√	√	√	√	√	√	√
GW191113_071753	···	···	···	√	···	···	···	···	···	···	···	···	···
GW191126_115259	···	···	···	···	···	···	···	···	···	√	···	···	···
GW191127_050227	···	···	√	√	√	···	···	···	···	√	√	√	√
GW191129_134029	···	···	√	√	√	√	√	√	√	√	√	···	···
GW191204_110529	···	···	···	···	···	···	···	···	···	√	···	···	···
GW191204_171526	√	√	√	√	√	√	√	√	√	√	√	√	√
GW191215_223052	√	√	√	√	√	√	√	√	√	√	√	√	√
GW191219_163120	···	···	···	···	···	···	···	√	···	···	···	···	···
GW191222_033537	√	√	√	√	√	√	√	√	√	√	√	√	√
GW191230_180458	√	√	···	···	√	√	···	···	√	√	√	√	√
GW200115_042309	···	···	√	√	√	√	√	√	···	···	···	···	···
GW200128_022011	√	√	···	√	√	√	√	√	√	√	√	√	√
GW200129_065458	···	···	···	···	√	√	√	√	√	√	√	√	√
GW200202_154313	···	···	···	···	√	√	···	···	√	√	···	···	···
GW200208_130117	···	···	√	√	√	√	√	√	√	√	···	√	···
GW200208_222617	···	···	···	···	···	···	···	···	···	√	···	···	···
GW200209_085452	···	···	···	√	√	√	···	···	√	√	√	···	···
GW200210_092254	···	···	···	···	√	···	···	√	···	√	···	···	···

(Table continued)

TABLE VI. *(Continued)*

	cWB		MBTA		GstLAL		PyCBC		PyCBC_BBH		AI model		
Name	FAR	p_{astro}	FAR	p_{astro}	FAR	p_{astro}	FAR	p_{astro}	FAR	p_{astro}	Detection	Test	D&T
GW200216_220804	…	…	…	…	√	√	…	…	…	√	…	…	…
GW200219_094415	√	√	√	√	√	√	√	√	√	√	√	√	√
GW200220_061928	…	…	…	…	…	…	…	…	…	√	…	…	…
GW200220_124850	…	…	…	√	…	…	…	…	…	…	√	…	…
GW200224_222234	√	√	√	√	√	√	√	√	√	√	√	√	√
GW200225_060421	√	√	√	√	√	√	√	√	√	√	√	√	√
GW200306_093714	…	…	…	√	…	…	…	…	…	…	…	…	…
GW200308_173609	…	…	…	…	…	…	…	…	…	√	…	…	…
GW200311_115853	√	√	√	√	√	√	√	√	√	√	√	√	√
GW200316_215756	…	…	…	…	√	√	√	√	√	√	…	…	…
GW200322_091133	…	…	…	√	…	…	…	…	…	…	…	…	…
Detection_rate (%)	31.25	31.25	40.63	59.38	62.5	56.25	46.88	53.13	53.13	78.125	50	40.63	37.5

show the detection results of five detection pipelines (cWB, MBTA, GstLAL, PyCBC_broad, PyCBC_BBH) and our AI model. Among the five pipelines, cWB searches for transient signals without assuming a model template [6], MBTA, GstLAL, PyCBC_broad, PyCBC_BBH are matched filter pipelines.

In GWTC-2, LVK used a threshold of FAR < 2 per year to screen the confident events from candidates. In the GWTC-2.1, eight events with $p_{\text{astro}} > 0.5$ were added to the confident events. p_{astro} represents the probability of astrophysical origin, which uses both the signal rate in addition to the noise rate in order to determine the significance of events [5]. In the current work, we use the same threshold of FAR and p_{astro} to analyze the GW events. For the five famous pipelines, we count the events that meet the requirements of $p_{\text{astro}} > 0.5$ and FAR < 2 per year, respectively. For the AI model, the threshold of the detection network is set to 0.9. Such a threshold matches the requirement of FAR < 2 per year. We count the confident events of the AI model with FAR < 2 per year and the corresponding detection rates of different pipelines are shown at the end of the table (the events number denoted by √ divided by the total events number listed in the table).

In O3a, there are 23, 26, and 21 events are successfully detected by the detection model, envelope extraction model, and the whole AI model. In O3b, 16, 13, and 12 events are successfully detected by the detection model, envelope extraction model, and the whole AI model. All the events detected by cWB are successfully detected by the AI model. In addition the AI model detects eight more events. However, the AI model does not achieve the detection effect of any matched filtering pipeline. Considering the detection result of O1 and O2 events, we estimate that our AI model can achieve about 80% efficiency of matched filtering pipeline under the same threshold of FAR < 2 per year. As the same as the leading machine learning search algorithm in MLGWSC-1, the efficiency of our AI model is also lower than the MF based algorithms under the condition of very low FAR.

TABLE VII. Execution time for each stage of the AI model.

Stage	Evaluating time
Whitening	<10 ms
WP denoise	<0.2 ms
Detection	<0.1 ms
Envelope extraction	$<2 \times 12$ ms

D. Evaluating the execution time and the real-time performance

Based on one NVIDIA 3060 GPU, we use about 10 days to train the three DL models (one detection model and two envelope extraction models). Even though the training time of the three DL models is time consuming, the execution time is extremely fast once trained. For testing the real-time performance of the AI model, the test set is used to evaluate the execution time of each stage. The data preprocessing (whitening and WP denoising) is run on CPU while the DL models are run on GPU. The execution time of every stage for a sample is shown in Table VII. Although the data preprocessing running on the CPU reduces the efficiency of AI model, the whole running time (about 0.03 s) is still much less than $\frac{1}{8}$ s (sliding time step). In the future, the data preprocessing operation is also expected to be accelerated by the GPU to further improve the efficiency of the AI model.

V. CONCLUSION AND DISCUSSION

In this paper, we propose an AI model for GW signal recognition. The AI model contains three stages: GW detection, envelope extraction, and coalescence time testing. Three AI models are trained by a new training scheme. We find that this new training scheme can extremely increase the distinguishing ability of GW signal and

background noise. Moreover, the envelope extraction can be used to accurately estimate the coalescence time. For about 80% of the samples in the test data set, the coalescence time prediction error is less than 5 ms. This AI model is applied to all reported events by LVK in the first, second, and third observing runs (O1, O2, and O3). We find that the AI model can clearly identify all binary black hole merger events in O1 and O2 except GW170818. We have also applied the AI model to one whole month (August 2017) of data of O2. Although the AI model is trained by only O1 data, no false trigger happens in the one month detection. In the investigation of O3 events, we compare the detection results of five famous detection pipelines (cWB, MBTA, GstLAL, PyCBC_broad, and PyCBC_BBH) to that of our AI model. We find that our AI model identifies eight more GW events than cWB besides all events that are successfully identified by cWB. Our test results indicate that our AI model can be used in real-time GW data analysis with good confidence.

Due to the improved performance of the detectors [57], the sensitivity of aLIGO during the third observing run is greatly improved. The power spectrum density of the O3 background noise is much different from that of O1. For experimental purposes, we only use the O1 background noise to train our AI model. Even so, our test results are promising. This fact indicates the strong generalization ability of the AI model. The performance of our AI model may be better if the background noise of the O3 run is also taken into consideration for training.

In the testing stage, we propose an envelope extraction network. The test based on the output of the envelope extraction network improves the efficiency of the GW signal detection quite much. Beyond the GW testing stage of the proposed AI model, the envelope extraction network has many other potential usages. Here we introduce some promising potential usages that can be investigated in future.

The data collected by Advanced LIGO interferometer are dominated by pure noise and have less GW information. However, so far the AI based detection method treats the data segments at different time (with and without GW information) equally. The information of the extracted envelope can pick up the significant data segment. Ones can afterwards focus rapidly on regions of interest to perceive an image [58]. We can imitate the attention mechanism in human vision and design AI detection method that focuses on the important time segments.

Almost all the GW detection methods based on AI use a time sliding method with a small sliding time step. The coalescence time information can be used to align the detection window. Then a new method without the small sliding time step can be applied. An aligned calculation window may also improve the effectiveness of the parameter estimation based on AI method. The coalescence time information may also be used to improve the computational efficiency of the matched filtering method.

ACKNOWLEDGMENTS

This research has made use of data and web tools obtained from the gravitational-wave Open Science Center, a service of LIGO Laboratory, the LIGO Scientific Collaboration, and the Virgo Collaboration. We thank Xinyao Yu for his comments on the manuscript. This work was supported in part by the National Key Research and Development Program of China Grant No. 2021YFC2203001, in part by the NSFC (Grants No. 11920101003 and No. 12021003) and the Natural Science Foundation of Jiangxi (Grant No. 20224BAB211012). Z. C. was supported by CAS Project for Young Scientists in Basic Research YSBR-006.

C. M. and W. W. contributed equally to this work and should be considered as cofirst authors.

[1] A. Einstein, Approximative integration of the field equations of gravitation, Sitzungsber. Preuss. Akad. Wiss. Berlin (Math. Phys.) **1916**, 1 (1916).

[2] A. Einstein, The general theory of relativity, in *The Meaning of Relativity* (Springer, New York, 1922), pp. 54–75.

[3] B. P. Abbott, R. Abbott, T. Abbott, M. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. Adhikari *et al.*, Observation of Gravitational Waves from a Binary Black Hole Merger, Phys. Rev. Lett. **116**, 061102 (2016),

[4] B. Abbott, R. Abbott, T. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, R. Adhikari, V. Adya, C. Affeldt *et al.*, GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo During the First and Second Observing Runs, Phys. Rev. X **9**, 031040 (2019).

[5] R. Abbott, T. Abbott, F. Acernese, K. Ackley, C. Adams, N. Adhikari, R. Adhikari, V. Adya, C. Affeldt, D. Agarwal *et al.*, Gwtc-2.1: Deep extended catalog of compact binary coalescences observed by LIGO and Virgo during the first half of the third observing run, arXiv:2108.01045.

[6] R. Abbott, T. Abbott, F. Acernese, K. Ackley, C. Adams, N. Adhikari, R. Adhikari, V. Adya, C. Affeldt, D. Agarwal *et al.*, GWTC-3: Compact binary coalescences observed by LIGO and Virgo during the second part of the third observing run, arXiv:2111.03606.

[7] B. P. Abbott *et al.*, Multi-messenger observations of a binary neutron star merger, Astrophys. J. Lett. **848**, L12 (2017).

[8] K. S. Thorne, Gravitational waves, arXiv:gr-qc/9506086.
[9] B. P. Abbott, R. Abbott, T. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. Adhikari, V. Adya *et al.*, GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral, Phys. Rev. Lett. **119**, 161101 (2017).
[10] B. P. Abbott, R. Abbott, T. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. Adhikari, V. Adya *et al.*, Gravitational waves and gamma-rays from a binary neutron star merger: GW170817 and GRB 170817A, Astrophys. J. Lett. **848**, L13 (2017).
[11] D. George and E. A. Huerta, Deep Learning for real-time gravitational wave detection and parameter estimation: Results with Advanced LIGO data, Phys. Lett. B **778**, 64 (2018).
[12] H. Gabbard, M. Williams, F. Hayes, and C. Messenger, Matching Matched Filtering with Deep Networks for Gravitational-Wave Astronomy, Phys. Rev. Lett. **120**, 141103 (2018).
[13] J. Redmon and A. Farhadi, Yolov3: An incremental improvement, arXiv:1804.02767.
[14] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, Feature pyramid networks for object detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, Honolulu, HI, 2017), pp. 2117–2125.
[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, Focal loss for dense object detection, in *Proceedings of the IEEE International Conference on Computer Vision* (IEEE, Venice, Italy, 2017), pp. 2980–2988.
[16] Y. Bengio, R. Ducharme, and P. Vincent, A neural probabilistic language model, Adv. Neural Inf. Process. Syst. **13**, 1137 (2000), https://dl.acm.org/doi/10.5555/944919.944966.
[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. **30**, 5998 (2017), https://dl.acm.org/doi/10.5555/3295222.3295349.
[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv:1810.04805.
[19] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, Deep learning in remote sensing: A comprehensive review and list of resources, IEEE Geosci. Remote Sens. Mag. **5**, 8 (2017).
[20] L. Zhang, L. Zhang, and B. Du, Deep learning for remote sensing data: A technical tutorial on the state of the art, IEEE Geosci. Remote Sens. Mag. **4**, 22 (2016).
[21] M. Liang, Q. He, X. Yu, H. Wang, Z. Meng, and L. Jiao, A dual multi-head contextual attention network for hyperspectral image classification, Remote sens. **14**, 3091 (2022).
[22] T. D. Gebhard, N. Kilbertus, I. Harry, and B. Schölkopf, Convolutional neural networks: A magic bullet for gravitational-wave detection?, Phys. Rev. D **100**, 063015 (2019).
[23] H. Wang, S. Wu, Z. Cao, X. Liu, and J.-Y. Zhu, Gravitational-wave signal recognition of LIGO data by deep learning, Phys. Rev. D **101**, 104003 (2020).
[24] E. Huerta, A. Khan, X. Huang, M. Tian, M. Levental, R. Chard, W. Wei, M. Heflin, D. S. Katz, V. Kindratenko *et al.*, Accelerated, scalable and reproducible AI-driven gravitational wave detection, Nat. Astron. **5**, 1062 (2021).
[25] W. Wei, A. Khan, E. Huerta, X. Huang, and M. Tian, Deep learning ensemble for real-time gravitational wave detection of spinning binary black hole mergers, Phys. Lett. B **812**, 136029 (2021).
[26] H. Xia, L. Shao, J. Zhao, and Z. Cao, Improved deep learning techniques in gravitational-wave data analysis, Phys. Rev. D **103**, 024040 (2021).
[27] M. B. Schäfer and A. H. Nitz, From one to many: A deep learning coincident gravitational-wave search, Phys. Rev. D **105**, 043003 (2022).
[28] C. Ma, W. Wang, H. Wang, and Z. Cao, Ensemble of deep convolutional neural networks for real-time gravitational wave signal recognition, Phys. Rev. D **105**, 083013 (2022).
[29] M. B. Schäfer, O. c. v. Zelenka, A. H. Nitz, H. Wang, S. Wu, Z.-K. Guo, Z. Cao, Z. Ren, P. Nousi, N. Stergioulas *et al.*, First machine learning gravitational-wave search mock data challenge, Phys. Rev. D **107**, 023021 (2023).
[30] kaggle, https://www.kaggle.com/competitions/g2net-gravitational-wave-detection/.
[31] B. J. Owen and B. S. Sathyaprakash, Matched filtering of gravitational waves from inspiraling compact binaries: Computational cost and template placement, Phys. Rev. D **60**, 022002 (1999).
[32] J. Veitch, V. Raymond, B. Farr, W. Farr, P. Graff, S. Vitale, B. Aylott, K. Blackburn, N. Christensen, M. Coughlin *et al.*, Parameter estimation for compact binaries with ground-based gravitational-wave observations using the LALInference software library, Phys. Rev. D **91**, 042003 (2015).
[33] H. Shen, D. George, E. Huerta, and Z. Zhao, Denoising gravitational waves with enhanced deep recurrent denoising auto-encoders, in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (IEEE, Brighton, United Kingdom, 2019).
[34] W. Wei and E. Huerta, Gravitational wave denoising of binary black hole mergers with deep learning, Phys. Lett. B **800**, 135081 (2020).
[35] C. Chatterjee, L. Wen, F. Diakogiannis, and K. Vinsen, Extraction of binary black hole gravitational wave signals from detector data using deep learning, Phys. Rev. D **104**, 064046 (2021).
[36] B. Allen, χ 2 time-frequency discriminator for gravitational wave detection, Phys. Rev. D **71**, 062001 (2005).
[37] A. Bohé, L. Shao, A. Taracchini, A. Buonanno, S. Babak, I. W. Harry, I. Hinder, S. Ossokine, M. Pürrer, V. Raymond *et al.*, Improved effective-one-body model of spinning, nonprecessing binary black holes for the era of gravitational-wave astrophysics with advanced detectors, Phys. Rev. D **95**, 044028 (2017).
[38] C. Cutler and E. E. Flanagan, Gravitational waves from merging compact binaries: How accurately can one extract the binary's parameters from the inspiral waveform?, Phys. Rev. D **49**, 2658 (1994).
[39] S. A. Usman, A. H. Nitz, I. W. Harry, C. M. Biwer, D. A. Brown, M. Cabero, C. D. Capano, T. Dal Canton, T. Dent, S. Fairhurst *et al.*, The PyCBC search for gravitational waves from compact binary coalescence, Classical Quantum Gravity **33**, 215004 (2016).
[40] T. Adams, D. Buskulic, V. Germain, G. Guidi, F. Marion, M. Montani, B. Mours, F. Piergiovanni, and G. Wang, Low-latency analysis pipeline for compact binary coalescences in

the advanced gravitational wave detector era, Classical Quantum Gravity **33**, 175012 (2016).
[41] C. Messick, K. Blackburn, P. Brady, P. Brockill, K. Cannon, R. Cariou, S. Caudill, S. J. Chamberlin, J. D. Creighton, R. Everett *et al.*, Analysis framework for the prompt discovery of compact binary mergers in gravitational-wave data, Phys. Rev. D **95**, 042001 (2017).
[42] E. Avci and Z. H. Akpolat, Speech recognition using a wavelet packet adaptive network based fuzzy inference system, Expert Syst. Appl. **31**, 495 (2006).
[43] F. Safara, S. Doraisamy, A. Azman, A. Jantan, and A. R. A. Ramaiah, Multi-level basis selection of wavelet packet decomposition tree for heart sound classification, Comput. Biol. Med. **43**, 1407 (2013).
[44] A. Y. Hindi, M. O. Dwairi, and Z. A. AlQadi, Creating Human Speech Identifier using WPT, Int. J. Comput. Sci. Mobile Comput. **9**, 117 (2020).
[45] C. Garcia, G. Zikos, and G. Tziritas, Wavelet packet analysis for face recognition, Image Vision Comput. **18**, 289 (2000).
[46] A. Fathi and A. R. Naghsh-Nilchi, Efficient image denoising method based on a new adaptive wavelet packet thresholding function, IEEE Trans. Image Process. **21**, 3981 (2012).
[47] M. Diwakar and M. Kumar, CT image denoising using NLM and correlation-based wavelet packet thresholding, IET Image Proc. **12**, 708 (2018).
[48] X.-R. Li, W.-L. Yu, X.-L. Fan, and G. J. Babu, Some optimizations on detecting gravitational wave using convolutional neural network, Front. Phys. **15**, 54501 (2020).
[49] B.-J. Lin, X.-R. Li, and W.-L. Yu, Binary neutron stars gravitational wave detection based on wavelet packet analysis and convolutional neural networks, Front. Phys. **15**, 24602 (2020).
[50] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv:1207.0580.
[51] M. B. Schäfer, O. Zelenka, A. H. Nitz, F. Ohme, and B. Brügmann, Training strategies for deep learning gravitational-wave searches, Phys. Rev. D **105**, 043002 (2022).
[52] C. E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. **27**, 379 (1948).
[53] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.
[54] O. Ronneberger, P. Fischer, and T. Brox, U-net: Convolutional networks for biomedical image segmentation, in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention* (Springer, New York, 2015), pp. 234–241.
[55] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, Rethinking the inception architecture for computer vision, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, Las Vegas, NV, USA, 2016), pp. 2818–2826.
[56] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (O'Reilly Media, 2019).
[57] A. Buikema, C. Cahillane, G. Mansell, C. Blair, R. Abbott, C. Adams, R. Adhikari, A. Ananyeva, S. Appert, K. Arai *et al.*, Sensitivity and performance of the Advanced LIGO detectors in the third observing run, Phys. Rev. D **102**, 062003 (2020).
[58] M.-H. Guo, T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R. R. Martin, M.-M. Cheng, and S.-M. Hu, Attention mechanisms in computer vision: A survey, in *Computational Visual Media* (Tsinghua University Press, Beijing, China, 2022), pp. 1–38.