2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

Improving Gravitational Wave Detection with 2D Convolutional Neural Networks

Siyu Fan* Yisen Wang† Yuan Luo* Alexander Schmitt‡ Shenghua Yu§
*Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
†Key Lab. of Machine Perception (MoE), School of EECS
Peking University, Beijing , China
‡Faculty of Economics and Business (FEB)
Katholieke Universiteit Leuven, Brussels , Belgium
§Joint Laboratory for Radio Astronomy Technology, National Astronomical Observatories
Chinese Academy of Sciences, Beijing, China
fansiyu@sjtu.edu.cn, yisen.wang@pku.edu.cn, luoyuan@cs.sjtu.edu.cn
info@a-schmitt.com, shenghuayu@bao.ac.cn

2020 25th International Conference on Pattern Recognition (ICPR) | 978-1-7281-8808-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPR48806.2021.9412180

***Abstract*—Sensitive gravitational wave (GW) detectors such as that of Laser Interferometer Gravitational-wave Observatory (LIGO) realize the direct observation of GW signals that confirm Einstein's general theory of relativity. However, it remains challenges to quickly detect faint GW signals from a large number of time series with background noise under unknown probability distributions. Traditional methods such as matched-filtering in general assume Additive White Gaussian Noise (AWGN) and are far from being real-time due to its high computational complexity. To avoid these weaknesses, one-dimensional (1D) Convolutional Neural Networks (CNNs) are introduced to achieve fast online detection in milliseconds but do not have enough consideration on the trade-off between the frequency and time features, which will be revisited in this paper through data pre-processing and subsequent two-dimensional (2D) CNNs during offline training to improve the online detection sensitivity.**

In this work, the input data is pre-processed to form a 2D spectrum by Short-time Fourier transform (STFT), where frequency features are extracted without learning. Then, carrying out two 1D convolutions across time and frequency axes respectively, and concatenating the time-amplitude and frequency-amplitude feature maps with equal proportion subsequently, the frequency and time features are treated equally as the input of our following two-dimensional CNNs. The simulation of our above ideas works on a generated data set with uniformly varying SNR (2-17), which combines the GW signal generated by PYCBC and the background noise sampled directly from LIGO. Satisfying the real-time online detection requirement without noise distribution assumption, the experiments of this paper demonstrate better performance on average compared to that of 1D CNNs, especially in the cases of lower SNR (4-9).

***Index Terms*—Signal analysis, Deep learning, Neural networks**

I. Introduction

Accurate gravitational wave detection which proves Einstein's general theory of relativity makes it possible to directly observe the collision of black holes and significantly facilitates the estimation of the exact mass and position of black holes. Thanks to the great development of the large-scale gravitational wave detection laser interferometers, such as the one led by Laser Interferometer Gravitational-wave Observatory (LIGO), capturing the gravitational wave is no longer infeasible. After the first detection of gravitational wave (GW) on September 14, 2015 [1], the new era of GW detection had been opened. Since then, several compact binary coalescence (CBC) observations [2]–[7] were detected one after another by Advanced LIGO [8] and Advanced Virgo [9] detectors.

However, the gravitational wave signals, which are produced very far away from the earth, are very faint, mixed with noise, and appear rather infrequently (the captured signals are pure noise mostly). Therefore, a sensitive detector and an effective data analysis method are both indispensable.

As interferometers, LIGO and VIRGO could serve as very strong infrastructures for gravitational wave capturing due to their high sensitivity. However, these sensitive detectors are also sensitive to the signals other than gravitational waves, such as trucks, lightning strikes, earthquakes, which may potentially lead to false alarms. A trivial but effective solution is to build a network of gravitational wave detectors (such as LIGO's two detectors located in Hanford and Livingston). If multiple detectors report positive within a certain time difference, it is considered to be a true gravitational wave event. However, building interferometers is extremely costly, so in this paper, we focus on improving the GW detection at the algorithm level.

Traditional methods such as matched-filtering have been proved to be successful in GW detection [10]–[14], whose key idea is to cross-correlate the signal templates (known signals) and the observation result (unknown signals) based on the Additive White Gaussian Noise (AWGN) assumption. However, this line of work suffers from inefficiency and is far from being real-time due to its high computational complexity. Therefore, detectors with higher speed and better generalization ability are needed.

Convolutional neural networks (CNNs) have demonstrated their effectiveness in the field of computer vision, in particular, image classification, and are fully loaded with the capacity of learning complex features when training data is sufficient.

Even though CNNs usually take more time in training, during inference time, they are quite fast (indeed, much faster then template-based methods). All the aforementioned merits make CNNs one of the best candidates to make a difference in the GW detection. Therefore, [15] and [16] are among the first attempts which apply CNNs in detecting gravitational wave. They regard the GW detection task as a classification problem. Specifically, they feed the raw time series obtained by the LIGO detectors into a CNN model, which performs several one-dimensional (1D) convolutions and outputs a label indicating whether the input data contains the GW signal or not. However, their models have not taken full advantage of the Convolution neural networks since CNNs are more effective at two-dimensional (2D) inputs. Furthermore, 1D time series itself is not condensed enough along the time dimension.

To address the limitations, in this paper, we propose to transform the original 1D time series into 2D spectrum with time and frequency dimensions using the Short-time Fourier transform (STFT). Considering the physical meanings of the two input dimensions are not comparable, we first perform 1D convolutions on each dimension individually at lower stages, and then concatenate them before using 2D convolutions at higher stages.

Note that, since none of the previous works released their datasets, all the experiments in this paper evaluate on the dataset generated according to the instructions of PYCBC [17]. Therefore, the positive samples in the dataset of this paper with SNR uniformly distributed from 2 to 17 may be different from the previous works, e.g. [15] and [16]. Please refer to Sec. IV-A for more details.

The main contributions of this paper include:

- **Feature Balance.** This paper firstly introduces 2D CNN to explore the GW detection in deep learning for performance improvement while keeping the balance of time-amplitude features and frequency-amplitude features. In fact, to enrich the feature of frequency without learning in data pre-processing, the raw time series is transformed to 2D spectrum through Short-time Fourier transform. And then, 1D convolutions are carried out along time and frequency axes of the spectrum respectively. Subsequently, with equal proportion, the time-amplitude and frequency-amplitude feature maps are concatenated, and then are finally fed into the following 2D convolutions.
- **Sensitivity Performance under Real Noise.** The proposed method averagely outperforms its 1D CNN counterparts over current test set with uniformly distributed SNR (2-17), whose sensitivity is improved from 75.9% to 77.8%. Furthermore, our model performs even better at lower SNR (4-9), for example, the increment is about 8% when SNR is around 7. Note that, the dataset of this paper chooses the background noise sampled directly from LIGO detectors instead of the synthetic Gaussian white noise, which leads to a more practical analysis.
- **Real-time Detection.** Compared with matched filtering, our method achieves the real-time online requirement without noise distribution assumptions. An eight-second time series can be processed online within milliseconds while the model parameters are trained and obtained offline. Meanwhile, our model has comparable speed with the method of 1D CNN.

The rest of this paper is organized as follows. Sec. II will introduce the related work of this work. In Sec. III, the problem definition, data pre-processing pipeline, and details of our network architecture are covered. Comprehensive experiments are reported in Sec. IV, which include the data generation, evaluation metric, the comparison among the performance of various methods, and ablation studies. Finally, Sec. V will briefly conclude this paper.

II. Related Work

Matched-filtering. Matched-filtering is a commonly used method in signal detection [18]. It is wildly applied in fields of radar [19], digital communication [20], image processing [21], [22], and gravitational-wave astronomy. Indeed, the first observed gravitational wave event GW150914 [1] was detected using matched-filtering methods [12]. Matching filtering requires the establishment of a reasonable physical model for the gravitational wave source, it then generates thousands of templates based on the model, and finally uses these templates to search for the relevant gravitational wave signals. However, since the high computational complexity due to large parameter space, matched-filtering is far from being real-time. Meanwhile, the GW detection task inherently demands high processing speed because the LIGO detectors keep collecting data in a streaming way, so it is preferred that there is no delay between data collecting and processing. In addition, the generalization ability of matched-filtering is limited by the templates, that is, it cannot detect the GWs which are not in the template library.

Convolutional Neural Network. Convolutional Neural Network (CNN) is a branch of deep neural networks, which is first proposed by LeCun for handwritten digital recognition [23]. It is widely used in pattern recognition fields [23]–[28].

The local connection of the network and the parameter sharing of the convolution kernels greatly reduce the number of parameters of the model and make the inference fast. With the rapid development of hardware equipment, CNN has become deeper and deeper. From the earliest LeNet5 [23] to later AlexNet [24], VGG-Net [26], GoogLeNet [27] and Resnet [28], the accuracy of the CNN is constantly improving.

The powerful feature extraction and generalization capability of CNN make it dominant in digital signal processing. Except for conventional 2D data such as images, CNN can also deal with 1D data [29] such as gravitational wave detection.

Gravitational Wave Detection With CNN. [15] is the first to apply CNN in GW detection which considers GW detection as a binary classification problem. They use simulated GW signals generated by PYCBC and synthesized background noise (Gaussian white noise) to simulate the time series received by LIGO, then feed the time-series data into a 1D CNN model. Later [30] uses the real LIGO data instead of synthesized noise with the same CNN architecture in the GW

parameter estimation task. [31] proves that the 1D CNNs can achieve the same performance of the matched-filtering on the same dataset. [16] feeds the real LIGO data into a 1D CNN, not only to predict the existence of the GW signal but also to determine the occurrence time of the GW event. Since most of the work focuses on leveraging 1D CNN in GW detection, whereas the CNN is known for its great capacity for 2D inputs, we propose to convert GW signals into 2D representation and classify them with a 2D CNN. According to our knowledge, we are the first to adopt 2D CNN in GW detection. And the novel 1D-then-2D design of our network makes it achieve state-of-the-art performance with comparable speed cost in contrast with other deep learning methods.

III. Method

In this section, the problem is formally defined in Sec. III-A, then Sec. III-B introduces the data preparation that converts 1D time-series data into 2D spectrum consisting of the components "time, frequency and amplitude" by Short-time Fourier transform (STFT). Different from the earlier works of [15] and [16], more frequency features are extracted without learning during STFT. Finally, Sec. III-C explores the architecture of our two-dimensional CNN model in three steps. The first step carries out two 1D convolutions across time and frequency axes respectively. Then the second step provides an opportunity to prevent unfair feature adoption by combining the time-amplitude and frequency-amplitude feature maps with equal proportion. By using this reconstructed data as the input of the 2D convolutions in the third step, two-dimensions of the data are treated equally in the neural network.

A. Problem Definition

Following [15], we consider the GW detection task as a binary classification task, where the positive samples are the time-series data containing GW signals and the negative samples are pure noise strains. Specifically, negative samples are real background noise collected by LIGO detectors and and the positive samples $s(t)$ are mixture of stimulated GW signals $h(t)$ and real background noise $n(t)$ as shown in (1):

$$s(t) = n(t) + h(t). \tag{1}$$

We use the signal-to-noise ratio (SNR) to represent the difficulty of recognizing positive samples, which indicates the ratio of the power of the target signal over that of background noise. And the smaller the SNR is, the more difficult to detect the GW signals as the background noise becomes more dominant. In this paper, we use the standard definition of optimal matched filter SNR, as described by [14], [32].

In evaluation, we consider the recall rate as the major metric and we also evaluate the speed overhead for practical purposes.

In conclusion, the goal of this paper is to propose a fast GW detection model that achieves a high recall rate across different levels of SNR. We divide the method illustration into two parts, namely data pre-processing and model construction as introduced below.

B. 1D Time-series data to 2D Spectrum

Each sample in our dataset is a pair of gravitational waves (GWs), which are obtained from two LIGO detectors observing the same signal. In addition, each GW is represented by a one-dimensional sequence whose only dimension denotes time. In practice, signals from the same pair are usually stacked together by aligning the time dimension T. Thus, a second dimension is created, which we name as the channel dimension C. We formally describe the data sample as follows:

$$g = [g_1; g_2] \in \mathbb{R}^{C \times T}, \tag{2}$$

where g_1 and g_2 denote the time series obtained from each LIGO detector respectively and $C = 2$. Each of the time series lasts for 8 seconds and is sampled at 2048 Hz, which results in $T = 16384$ sample points. As g only has one dimension, apart from the channel dimension, when directly using g as the input of the Convolutional Neural Networks (CNNs), 1D convolution becomes the only applicable choice. Specifically, in each convolutional layer, a kernel $K_{C \times W}$ slides along the time dimension while performing the dot product with the input of the current layer, where the parameter W denotes the kernel size.

While leveraging the 1D CNN for GW detection is intuitive, as the previous work [15], [16] do, it remains a sub-optimal choice compared to 2D CNNs because (1) 1D data is less representative than 2D data, which overlook the frequency features; (2) 2D CNNs are well-developed in the field of image processing and CNNs are proven to be extremely good at dealing with 2D inputs; (3) the time dimension $T = 16384$ is large but sparse, therefore actions need to be taken to condense the time dimension.

Thus, we propose to use the Short-time Fourier transform (STFT) to convert 1D GW signals into 2D ones, and then feed them into a 2D CNN which by design considers the characteristics of the STFT spectrum.

Specifically, STFT first breaks the whole signal into small pieces and then performs FFT on each sub-sample. In this way, each sub-sample captures the frequency information within a small time window. And by concatenating all the transferred sub-samples, the time dimension can be reconstructed. The length of the sub-sample is a hyper-parameter named window size w. A larger window size makes the Fourier transform more accurate (better frequency feature) while a smaller window size preserves more time-axis features. Note that, the sub-samples can overlap with their neighbors. As the overlapping ratio (the intersection of two pieces divided by their union) increases, the information along the frequency dimension becomes more redundant. While the redundancy encourages the robustness, it also hurts the computational efficiency. This trade-off is modeled with a second hyper-parameter termed stride s, which denotes the distance between the centers of two sub-samples next to each other. Mathematically, the SFTF functions we use are as follows:

$$G_1[m,k] = \sum_{n=0}^{T-1} g_1[n] w[n-m] e^{-j\frac{2\pi}{T}kn}, \tag{3}$$

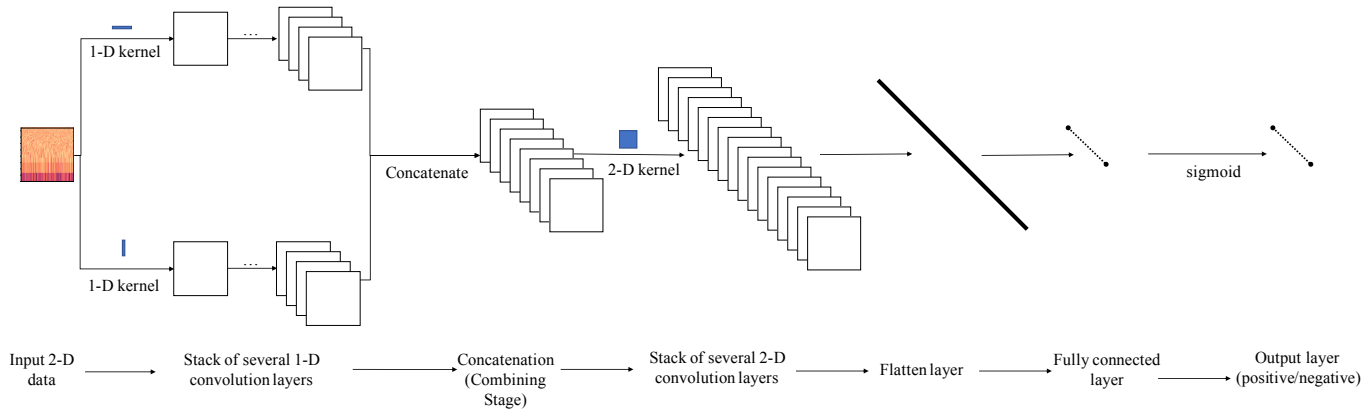


Fig. 1: **Model architecture.** The input of the model is 2D spectrum. The model first performs several 1D convolutions along the time and frequency dimensions separately and then concatenates the two types of feature along the channel axis before using 2D convolutions. Finally, the feature is flattened and goes through the fully connected layer to obtain the probability.

$$G_2[m,k] = \sum_{n=0}^{T-1} g_2[n]w[n-m]e^{-j\frac{2\pi}{T}kn}, \tag{4}$$

where j is imaginary unit, k represents the sample index in the frequency series, $w[\cdot]$ denotes the window function, $g_1[\cdot]$ and $g_2[\cdot]$ are time series denoted in (2), $G_1[\cdot]$ and $G_2[\cdot]$ are frequency representations of the windowed data centered at time m. Note that the window function defines how to chop the whole signal into pieces. And to reduce the artifact of the chopping, options such as Hann window and Gaussian window are usually chosen. This paper empirically select the Hann window. In addition, as mentioned before, the choices of the window size and stride are also critical to optimal performance. Following the common practice, stride is set to half of the window size. In addition, we also perform comprehensive experiments on the various window sizes and finalize the choice according to the results. As later covered in Sec. IV, adapting STFT into the GW detection task is non-trivial as the performance fluctuates with the different combinations of hyper-parameters. And we hope our work can contribute to the future work in converting 1D GW signals to 2D STFT spectrum.

C. Model Architecture

After the data pre-processing of STFT, the one-dimensional time series are converted into 2D time-frequency data, which is denoted as $G = \{(G_1[m,k]; G_2[m,k])\} \in \mathbb{R}^{C\times T'\times F}$, where T' is the length of time (m) feature and F is the length of frequency (k) feature, $C = 2$, as we stack $G_1[m,k]$ and $G_2[m,k]$. The Gravitational Wave detection task is treated as a binary classification problem and leverage the powerful convolutional neural networks to obtain a data-driven classifier $f(\cdot)$. Each sample in the training dataset is an instance of G which is denoted by $G^{(i)}$ $(i = 1,\ldots,N)$ with corresponding ground-truth label $l^{(i)}$. The label distinguishes the spectrum containing GW signals (positive samples) from pure noise (negative samples). The goal of our CNN classifier is to minimize the loss function $c(\cdot)$ in data training and to accurately predict the correct label in data testing:

$$\hat{\theta} = \underset{\theta}{argmin} \sum_{i=1}^{N} c(f_\theta(G^{(i)}), l^{(i)}), \tag{5}$$

where θ denotes the parameters of the CNN model which is to be learned, and the training dataset contains N samples.

Even though the 2D STFT spectrum input makes the usage of 2D CNN models possible, directly borrowing the state-of-the-art image classifiers, such as ResNet [28], into the GW detection task fails to produce satisfactory results. The key reason, as we observe, is that the 2D convolution kernel, which essentially is a sliding weighted sum operation, does not distinguish the two dimensions in the input feature. While this works as expected when inputs are image features since the two dimensions, which are height and width, can be treated equally, this raises a problem for the STFT feature because the physical meanings of the two dimensions are time and frequency which are not comparable at all. In conclusion, it is not reasonable to directly apply two-dimensional convolution kernels on the STFT feature.

Fortunately, the CNN models are consist of multiple convolution layers with activation layers in between to achieve non-linearity. The typical interpretation of the mechanism of CNN is that lower layers (close to the input) capture the low-level information and the feature output from higher layers contains more semantic information by aggregating and summarizing the low-level information with a global view. Moreover, as the feature becomes more abstract when going through multiple low-level layers, it gradually loses its pre-defined physical meaning. Thus, we propose that, in lower layers, the time and frequency axes should be treated discriminately and in higher layers, it is reasonable to operate the two axes equally. To this end, we apply two sets of 1D convolution layers in the lower stage, where convolution layers from different sets are totally independent. Specifically, convolution kernels from two sets slide along the time and frequency dimension respectively,

resulting in time-major and frequency-major feature. We then concatenate the two types of features along the channel axis at the combining stage, and use the 2D convolution kernels on the concatenated feature. After the feature going through multiple 2D convolutions, we follow the classic CNN design for the classification task, which is flattening the feature, appending a fully connected layer, and using softmax function to generate the probability of the sample being positive or negative. Please refer to our network visualization for a better understanding. See Fig. 1.

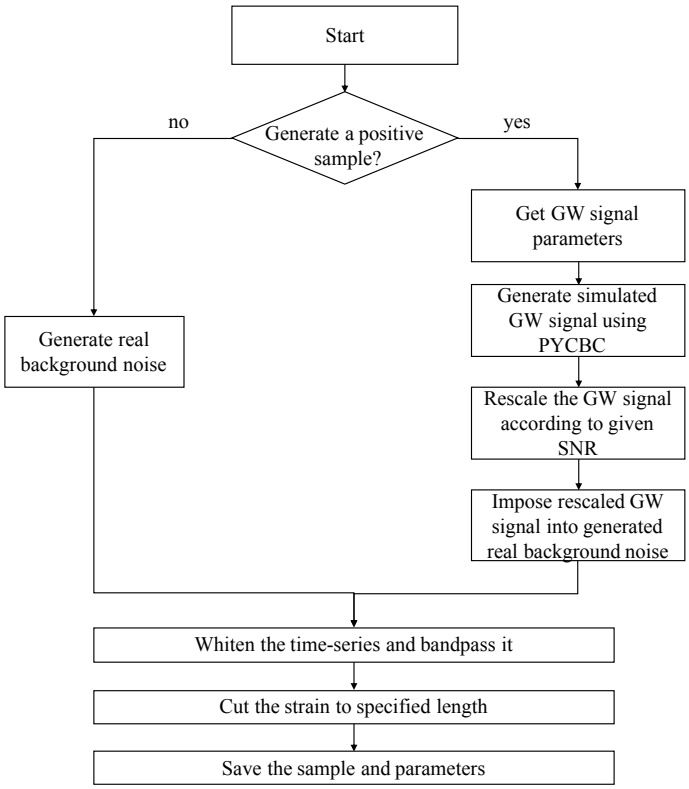


Fig. 2: The flowchart shows the process of generating a positive or negative sample.

IV. Experiments

In this section, the data generation process is first introduced in Sec. IV-A. Then evaluation metric for our experiments is demonstrated in Sec. IV-B. Next, the implementation details, such as model hyper-parameters, are shown in Sec. IV-C. Finally, the results of various models with different hyper-parameters are presented in Sec. IV-D. As shown in Table I, the overall performance of our 2D model is higher than that of the 1D CNN [15], especially when the SNR is low (4-9).

A. Data Generation

Our method, together with its alternatives, falls into the category of data-driven methods. Furthermore, because CNNs have much more parameters than the traditional machine learning algorithms, plentiful and diverse training data is necessary to prevent overfitting. However, throughout history, there are only several binary black hole coalescences being observed. Therefore, positive training samples are far from being sufficient. Fortunately, open-source packages, such as PYCBC, provide simulation algorithms to generate various gravitational waveforms. And in this paper, PYCBC is leveraged as the major tool for data generation purposes.

Our data generation process, as shown in Fig. 2, can be divided into several steps, namely background noise generation, GW simulation, mixing noise with GW, and post-processing.

1) **Background noise generation.** For simplification, previous works use synthetic white Gaussian noise as the background noise when generating training and testing samples. However, GW detectors trained with such datasets might not be directly applied under the real circumstance since synthetic noise does not contain transient noise such as glitches [33], which can not be ignored in practice. Therefore, to demonstrate the practical value and capacity of our model, all the background noise in our dataset is from the real open-source LIGO recordings.
2) **GW simulation.** Since positive samples (noise mixed with GW signal) are rather rare in the LIGO recordings, training a deep neural network solely on the real GW signals is destined to overfitting. Thus, we leverage the PYCBC tool to simulate GW signals given various combinations of hyper-parameters to ensure both the abundance and diversity of our training dataset. Specifically, we first generate two polarization modes (plus and cross polarization) of the simulated waveform with certain masses and spins. Then we project the two raw waveforms onto the antenna patterns of the detectors in Hanford and Livingston with certain hyper-parameters that control the position of the source in the sky such as right ascension, declination, and polarization angle.
3) **Mixing noise with GW.** To mimic the real scenario, where the input signals have a variety of SNRs, we rescale the amplitude of GW signals according to certain SNR before injecting them into the background noise. We choose the SNR for each sample by a uniform distribution ranging from 2 to 17 in order to encourage the robustness of our network.
4) **Post-processing.** In this step, we first whiten the time series and bandpass them to get rid of some non-physical turn-on artifacts introduced by the simulation. Next, we cut the time series into a fixed-length (8 seconds in this work) as demanded by our model. Finally, we record all the hyper-parameters that have been used to generate each sample, such as SNR.

In this work, each sample has two time series observed from two detectors in LIGO. Every time series is sampled at 2048 Hz with 8 seconds duration and the GW appears at 5.5s in the time series if the sample is positive. In total, our training set consists of 24576 positive samples and 8192 negative samples, our validation set contains 3072 positive samples and 1024 negative samples, and our testing set has

12288 positive samples and 4096 negative samples. All the datasets have the same hyper-parameter space: the mass of two binary black holes are uniformly distributed from 10 to 80, the z component of both binary component's dimensionless spin are uniformly distributed from 0 to 0.998, and the SNRs of the positive samples are uniformly distributed from 2 to 17. Note that all of our dataset splits have no overlap with each other.

B. Evaluation metric

In practice, the GW appears rather infrequently, therefore we cannot afford to miss any GW signals. Thus, sensitivity, or recall rate, can serve as an effective evaluation metric. To prevent the model from cheating by always outputting positive, we set up a fixed false alarm rate of 0.6%. In addition, to better compare with the previous work, we also evaluate the performance of our CNN model under different SNRs. Specifically, we split the test set into 15 bins, ranging from (2, 3) to (16, 17). Each bin contains around 1092 samples, of which positive samples account for $\frac{3}{4}$. We sort the predictions by the confidence in a descending manner and discard the rest predictions if false positives reach the max tolerance.

C. Implementation details

We have done various experiments with different combinations of the hyper-parameters of the model, including the number of convolutional layers, the combining stage (at which layer we combine the one-dimensional convolution intermediate data), kernel size and the number of kernels in every layer. We finally find out the following configurations to be the most effective (best performance on the test set):
The model contains two one-dimensional convolutional layers with 1×5 and 5×1 kernel size in two directions separately and two two-dimensional convolutional layers afterward, with the kernel size of 5×5. We set the stride and padding to 1 and 2 respectively to avoid information loss. And the depth for each layer (the number of kernels) is 64, 128, 256, and 512.

During training, we use the common binary Cross-Entropy as the loss function, which is a combination of log softmax and negative log likelihood loss. Adam [34] is used as the optimizer for iteratively updating the weights of the neural network based on the training loss. Compare to the traditional stochastic gradient descent (SGD) algorithm, we find that Adam produces better results and can be efficiently calculated with less memory.

After each training epoch, we calculate the loss on the validation set and update the learning rate (LR) according to the validation loss. We adopt the dynamic learning rate reducing scheduler implemented by PYTORCH named ReduceLROnPlateau, which reduces the LR by a factor of 0.5 if the validation loss has not gone down for at least 10 training epochs. We set the initial value of the LR to be 3×10^{-4} and once the LR is below 10^{-6}, we will consider the model to be converged and terminate the training process early. The total training epoch is 64 and the batch size is 64. Each training instance is launched on four 32G V100 graphics cards and the training typically takes around 2 hours.

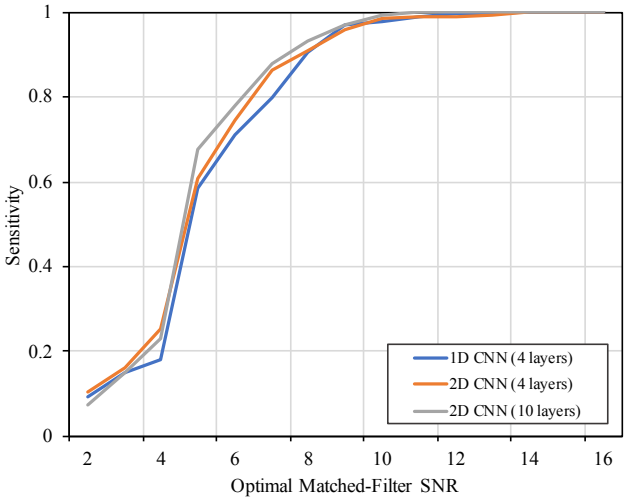


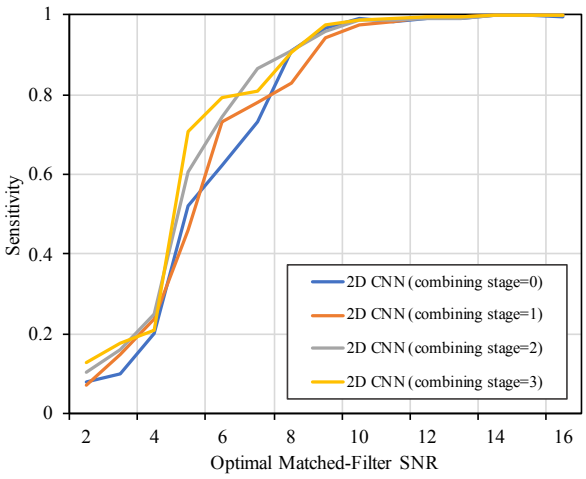
Fig. 3: **Comparison between 1D and 2D models within different SNR bins.** We split the test dataset according to different SNR ranges and compare 1D and 2D CNNs under different difficulty levels. The lower the SNR, the harder the task is. Our methods are particularly better in harder cases compared to 1D CNN.

Model (layers)	combining stage	Sensitivity (%)
1D CNN (4) [15]	N/A	75.9
2D CNN (4)	2	**77.8**
2D CNN (10)	4	77.7

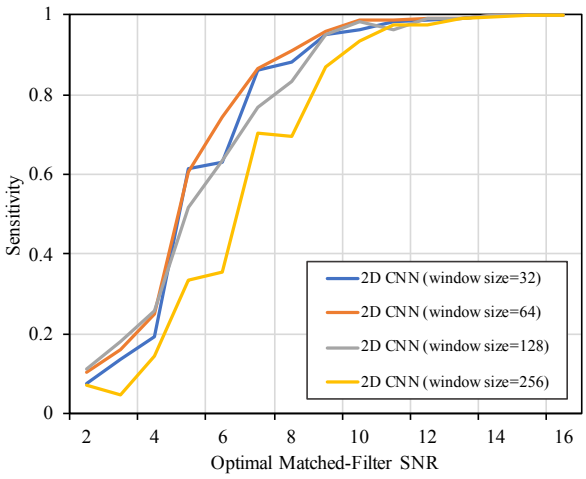
TABLE I: **Comparison between 1D and 2D models.** Here we calculate the overall sensitivity on the test dataset without splitting the data into different SNR bins. Note that for 2D models, STFT with window size of 64 is used in data preprocessing and the 1D CNN is re-implemented by us.

D. Results

Comparison with 1D CNN. In addition to the aforementioned model, we also develop a deeper 2D CNN with 10 convolutional layers whose combining stage is empirically set as 4 to further explore if a deeper network boosts the performance. As shown in Tab. I, both of our proposed models surpass the 1D CNN [15] by a margin of 1.9%. In addition, we split the test set according to the SNR to further evaluate the performance of different models, Fig. 3 shows that 2D models work particularly well in low SNR bins (harder cases), especially when SNR bin is (6,7), the sensitivity of the 2D model is about 8% higher than that of the 1D CNN. We think the performance gain mainly comes from separately modeling the time and frequency feature which makes the GW signal more discriminative. Note that 10-layer configuration has almost the identical performance with the 4-layer one. This may due to that the capacity of the 4-layer 2D CNN is already enough to handle our dataset. In other words, with the current dataset diversity, 10-layer 2D CNN potentially suffers from overfitting problems.



(a) Different combining stages



(b) Different STFT window sizes

Model	Combining stage	Sensitivity (%)
2D CNN	0	74.7
2D CNN	1	74.3
2D CNN	2	**77.8**
2D CNN	3	77.2

(c) Different combining stages

Model	Window size	Sensitivity (%)
2D CNN	32	76.2
2D CNN	64	**77.8**
2D CNN	128	74.8
2D CNN	256	68.1

(d) Different STFT window sizes

Fig. 4: **Ablation on combining stages and STFT window sizes.** All the experiments are done with the 4-layer 2D CNN configuration. In (a) and (b), we split the test dataset according to different SNR bins to show the performance breakdown within each SNR bin. In (c) and (d), we calculate the sensitivity across the whole dataset to show the overall performance of each setting. Note that in (a) and (c), we fix the window size to 64 while varying the combining stage. And in (b) and (d), the window size varies with the combining stage set to 2.

Ablation Study on combining stage. As we have mentioned before, our network first adopts 1D convolution layers along the time and frequency separately and then combines the feature together, after which point, 2D convolutions are applied. The transition layer is termed as the combining stage, for example, combining stage $= 0$ means the model has no 1D convolution layer and directly does four 2D convolutions after the input, combining stage $= 1$ means the model has one 1D convolution layer and three 2D convolution layers after the input. When the combining stage is too small (combine too early), the feature is not abstract enough and the two dimensions should not be treated equally, while a too-large combining stage decrease the effect of 2D convolutions. Thus, in Tab. 4c and Fig. 4a, we show all the choices of the combining stage and find that setting combining stage to 2 yields the best performance. Besides, when directly applying 2D convolutions to the spectrum, i.e. combining stage = 0, the performance degrades significantly, which proves the proposed 1D-then-2D model is necessary.

Ablation Study on STFT window size. STFT helps with converting the 1D time feature into the 2D time-frequency feature. Window size, as one of the hyper-parameters of STFT, controls the trade-off between having more time or frequency feature. A larger window size results in a more accurate frequency extraction but shrinks the time dimension more severely. In Tab. 4d and Fig. 4b, we experiment with a variety of window sizes with 64 as the best choice.

Speed Comparison. To demonstrate that our method improves the GW detection with a marginal speed cost, we compare the inference time of our models and 1D CNN [15]. As shown in Tab. II, our method has the same processing magnitude as 1D CNN and the major overhead comes from the STFT pre-processing.

Remark. Note that, even our method is almost twice slower, it can still finish processing the eight-second time series within 5 milliseconds, which far exceeds the real-time requirement. In addition, the real optimal solution matched filtering is far away from the real-time requirement, e.g. the PYCBC pipeline spent 530 CPU days [12].

Model	Pre-process (ms)	Process (ms)	Total (ms)
1D CNN (4 layers)	0.56	1.66	2.22
2D CNN (4 layers)	1.93	2.33	4.26
2D CNN (10 layers)	1.98	2.54	4.52

TABLE II: **Speed comparison between 1D and 2D CNNs.** *Pre-process* indicates the time spent on data loading and STFT if necessary. *Process* shows the network forwarding speed. The time statistics are averaged over the whole test dataset.

V. Conclusion

This paper presents a novel 2D convolutional neural network for gravitational wave detection. We pre-process the data through a Short-time Fourier transform which converts the 1D signal to the 2D spectrum. And the 2D CNN first differentiates the two dimensions of the 2D spectrum and treats them equally later in higher stages. Experiments and analysis show the state-of-the-art performance of our model on the real LIGO data with comparable time cost to 1D CNNs.

Acknowledgment

Special thanks to Dr. Frank Ohme from Max Planck Institute for Gravitational Physics, Hanover, for discussing the current progress of LIGO and how the research could make an impact on the current situation. We also would like to thank Professor Dr. Sebastiano Bernuzzi from the Friedrich Schiller University in Jena for discussing the current progress in LIGO GW detection and its potential.

This work was supported in part by China Program of International S&T Cooperation 2016YFE0100300.

References

[1] B. P. Abbott, R. Abbott, T. Abbott *et al.*, "Observation of gravitational waves from a binary black hole merger," *Physical review letters*, vol. 116, no. 6, p. 061102, 2016.

[2] B. P. Abbott, R. Abbott *et al.*, "Gw151226: observation of gravitational waves from a 22-solar-mass binary black hole coalescence," *Physical review letters*, vol. 116, no. 24, p. 241103, 2016.

[3] L. Scientific, B. Abbott, R. Abbott, T. Abbott *et al.*, "Gw170104: observation of a 50-solar-mass binary black hole coalescence at redshift 0.2," *Physical Review Letters*, vol. 118, no. 22, p. 221101, 2017.

[4] B. P. Abbott, R. Abbott, T. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. Adhikari, V. Adya *et al.*, "Gw170608: Observation of a 19 solar-mass binary black hole coalescence," *The Astrophysical Journal Letters*, vol. 851, no. 2, p. L35, 2017.

[5] B. P. Abbott, R. Abbott, T. Abbott *et al.*, "Gw170814: a three-detector observation of gravitational waves from a binary black hole coalescence," *Physical review letters*, vol. 119, no. 14, p. 141101, 2017.

[6] B. Abbott, R. Abbott, T. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, R. Adhikari, V. Adya, C. Affeldt *et al.*, "Gw190425: Observation of a compact binary coalescence with total mass $\sim$ 3.4m," *arXiv preprint arXiv:2001.01761*, 2020.

[7] R. Abbott, T. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, R. Adhikari, V. Adya, C. Affeldt, M. Agathos *et al.*, "Gw190412: Observation of a binary-black-hole coalescence with asymmetric masses," *arXiv preprint arXiv:2004.08342*, 2020.

[8] J. Aasi, B. Abbott, R. Abbott, T. Abbott *et al.*, "Advanced ligo," *Classical and quantum gravity*, vol. 32, no. 7, p. 074001, 2015.

[9] G. M. Harry, L. S. Collaboration *et al.*, "Advanced ligo: the next generation of gravitational wave detectors," *Classical and Quantum Gravity*, vol. 27, no. 8, p. 084006, 2010.

[10] B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. Creighton, "Findchirp: An algorithm for detection of gravitational waves from inspiraling compact binaries," *Physical Review D*, vol. 85, no. 12, p. 122006, 2012.

[11] S. Babak, R. Biswas, P. Brady, D. A. Brown, K. Cannon, C. D. Capano, J. H. Clayton, T. Cokelaer, J. D. Creighton, T. Dent *et al.*, "Searching for gravitational waves from binary coalescence," *Physical Review D*, vol. 87, no. 2, p. 024033, 2013.

[12] 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 and Quantum Gravity*, vol. 33, no. 21, p. 215004, 2016.

[13] 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," *Physical Review D*, vol. 95, no. 4, p. 042001, 2017.

[14] B. J. Owen and B. S. Sathyaprakash, "Matched filtering of gravitational waves from inspiraling compact binaries: Computational cost and template placement," *Physical Review D*, vol. 60, no. 2, p. 022002, 1999.

[15] D. George and E. Huerta, "Deep neural networks to enable real-time multimessenger astrophysics," *Physical Review D*, vol. 97, no. 4, p. 044039, 2018.

[16] T. D. Gebhard, N. Kilbertus, I. Harry, and B. Schölkopf, "Convolutional neural networks: a magic bullet for gravitational-wave detection?" *Physical Review D*, vol. 100, no. 6, p. 063015, 2019.

[17] N. Alex, H. Ian, B. Duncan, M. B. Christopher *et al.*, "gwastro/pycbc: Pycbc release 1.16.4 (version v1.16.4)." Jun. 2020, http://doi.org/10.5281/zenodo.3904502.

[18] G. Turin, "An introduction to matched filters," *IRE transactions on Information theory*, vol. 6, no. 3, pp. 311–329, 1960.

[19] J. R. Roman, M. Rangaswamy, D. W. Davis, Q. Zhang, B. Himed, and J. H. Michels, "Parametric adaptive matched filter for airborne radar applications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 2, pp. 677–692, 2000.

[20] B. B. Ibrahim and A. H. Aghvami, "Direct sequence spread spectrum matched filter acquisition in frequency-selective rayleigh fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 5, pp. 885–890, 1994.

[21] A. A.-H. A.-R. Youssif, A. Z. Ghalwash, and A. A. S. A.-R. Ghoneim, "Optic disc detection from normalized digital fundus images by means of a vessels' direction matched filter," *IEEE transactions on medical imaging*, vol. 27, no. 1, pp. 11–18, 2007.

[22] A. Hoover, V. Kouznetsova, and M. Goldbaum, "Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response," *IEEE Transactions on Medical imaging*, vol. 19, no. 3, pp. 203–210, 2000.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[25] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[29] S. Kiranyaz, T. Ince, O. Abdeljaber, O. Avci, and M. Gabbouj, "1-d convolutional neural networks for signal processing applications," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8360–8364.

[30] D. George and E. Huerta, "Deep learning for real-time gravitational wave detection and parameter estimation: Results with advanced ligo data," *Physics Letters B*, vol. 778, pp. 64–70, 2018.

[31] H. Gabbard, M. Williams, F. Hayes, and C. Messenger, "Matching matched filtering with deep networks for gravitational-wave astronomy," *Physical review letters*, vol. 120, no. 14, p. 141103, 2018.

[32] 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?" *Physical Review D*, vol. 49, no. 6, p. 2658, 1994.

[33] M. Zevin, S. Coughlin, S. Bahaadini, E. Besler, N. Rohani, S. Allen, M. Cabero, K. Crowston, A. K. Katsaggelos, S. L. Larson *et al.*, "Gravity spy: integrating advanced ligo detector characterization, machine learning, and citizen science," *Classical and Quantum Gravity*, vol. 34, no. 6, p. 064003, 2017.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.