

Novel deep learning approach to detecting binary black hole mergers

Damon Beveridge,^{1,*} Alistair McLeod,^{1,†} Linqing Wen,^{1,‡} and Andreas Wicenec²

¹*Department of Physics, The University of Western Australia,
35 Stirling Highway, Crawley, WA 6009, Australia*

²*International Centre for Radio Astronomy Research,
The University of Western Australia M468, 35 Stirling Hwy, Crawley, WA 6009, Australia*
(Dated: January 15, 2025)

Gravitational wave detection has opened up new avenues for exploring and understanding some of the fundamental principles of the Universe. The optimal method for detecting modeled gravitational-wave events involves template-based matched filtering and performing a multidetector coincidence search in the resulting signal-to-noise ratio time series. In recent years, advancements in machine learning and deep learning have led to a flurry of research into using these techniques to replace matched filtering searches and for efficient and robust parameter estimation of the gravitational wave sources. This paper presents a feasibility study for a novel approach to detecting binary black hole gravitational wave signals, which utilizes deep learning techniques on the signal-to-noise ratio time series produced from matched filtering. We show that a deep-learning search can efficiently detect binary black hole gravitational waves from the signal-to-noise ratio time series in simulated Gaussian noise with simulated transient glitches. Furthermore, our search method can outperform a maximum SNR-based matched filtering search on simulated data of the Hanford and Livingston LIGO detectors in the presence of glitches. Lastly, since we are building upon the foundations of a matched filtering search pipeline, we can extract estimates for the signal-to-noise ratio and detector frame chirp mass of a gravitational wave event with similar accuracy as existing pipelines.

I. INTRODUCTION

Gravitational waves (GWs) were first detected by the Laser Interferometer Gravitational-Wave Observatory (LIGO) [1] on September 14, 2015 [2]. Since then, GW events from compact binary coalescences (CBCs) have been detected regularly across three separate observing runs [3–6] by the LIGO and Virgo [7] detectors, and the fourth observing run is now underway with the addition of the Kamioka Gravitational Wave Detector (KAGRA) [8]. With upgrades to these existing detectors improving their sensitivity and additional detectors such as LIGO-India [9] coming online, the opportunity to detect new GW events is increasing dramatically with each observing run. An increase in sensitivity brings an expected increase in the number of noise transients and complexity of gravitational wave data, making the development of sensitive search algorithms for the future of gravitational wave astronomy more important.

Currently, four CBC search pipelines operate in real-time to detect modeled gravitational-wave events. These pipelines are PyCBC Live [10, 11], GstLAL [12, 13], SPIIR [14] and MBTA [15] and are all based on the technique of matched filtering, which employs a bank of modeled GW waveform templates to perform a cross-correlation with the detector strain data to identify matching signals. When the output from matched filtering achieves a predefined threshold, together with a signal consistency test [16], and no data quality issues

arise [17], an event trigger can be produced. These tests, however, are not always robust in the presence of transient glitches in the detector noise and can still produce significant triggers without the presence of a gravitational wave [18, 19], or miss a detection due to the presence of a glitch [20].

Deep learning [21–24] is a subset of machine learning that uses artificial neural network models to make predictions by extracting features from input data for problems such as classification and regression. In recent years, machine learning and deep learning have been applied to gravitational-wave physics for problems such as binary black hole (BBH) merger detection [25–74], glitch identification and classification [65, 75–87], and parameter estimation [29, 51, 57, 88–95], with promising results.

Previous efforts on detecting gravitational waves from BBH mergers with deep learning has looked at making detections from the detector strain data [25–54] or time-frequency spectrograms [54–70]. It has been demonstrated that deep learning models can reach a similar sensitivity or even outperform the current pipelines for BBH mergers with a source frame chirp mass (see Eq. 3) greater than $\sim 15 M_{\odot}$ [46, 48]. However, detecting CBC mergers from these data sources using deep learning becomes difficult when looking at lower-mass events such as binary neutron star (BNS) and neutron star-black hole (NSBH) mergers, as well as low-mass BBH mergers [46, 48]. This arises due to the GW signal duration increasing for lower-mass binaries, meaning the signal power is spread over a longer period, resulting in a lower overall amplitude in the detector strain for a given signal-to-noise ratio (SNR). There have also been publications highlighting efforts to detect BBH mergers that use the peak values of the SNR time series produced by

* damon.beveridge@research.uwa.edu.au

† alistair.mcleod@research.uwa.edu.au

‡ linqing.wen@uwa.edu.au

matched filtering as input to a deep learning model, with small template banks consisting of 35 [71] and 57 [72] template waveforms. An additional challenge for this area of research is developing a robust search method in the presence of nonstationary and non-Gaussian noise, especially those in the form of transient detector glitches [46, 63, 74, 96–98].

This paper presents an alternative approach to detecting gravitational waves from binary black hole mergers using the SNR time series produced by matched filtering as the input to a deep learning model. This technique is advantageous in several aspects, including the fact that the SNR time series is readily available from current online and offline search pipelines, allowing an efficient implementation of this model to a current pipeline without additional data preprocessing and matched filtering computation. Another advantage is that the SNR time series projects the power of a gravitational wave signal into a small time window and is consistent for all CBC source types, allowing our deep learning model to be easily extended to detect lower mass mergers. Furthermore, matched filtering is the optimal search method for identifying modeled signals in stationary Gaussian noise [99]. However, data from the detectors can be non-stationary and contain transient glitches from environmental and instrumentation sources [100, 101]. In this paper, we present a feasibility study using this approach to detect BBH signals. We use simulated Gaussian data from the Hanford (H1) and Livingston (L1) LIGO detectors with injected glitches to show that this deep learning approach can effectively detect BBH signals, especially in glitchy detector noise. Lastly, because we build our search method on top of matched filtering, we show how we can produce estimates for the SNR and detector frame chirp mass of a detected gravitational wave signal with similar accuracy to existing matched filtering search pipelines, despite selecting the trigger associated with these values based on the output of the model instead of the network SNR.

Section II discusses the methods we have used to generate data to train and test our deep learning model, the architecture of the deep learning model and the search pipeline setup used to make detections. Section III presents the results of our search method in terms of robust false alarm rate assignment, detection sensitivity, and parameter recovery. Lastly, Sec. IV summarizes our findings and presents further research that could be done in this area, as well as future applications of this work.

II. METHOD

A. Matched filtering

Matched filtering is a fundamental technique in detecting gravitational waves and is crucial in identifying known signals buried in noisy detector data. It involves cross-correlating the detector strain, s , with a bank of

template waveforms, h . The output of the matched filtering operation forms the signal-to-noise ratio (SNR) time series, $\rho(t)$, which can be used to estimate the likelihood of a GW signal being present in the data. A template waveform, $h(\theta)$, is defined by its intrinsic source parameters θ , and the matched filtering operation is performed individually for each template waveform in the bank. The SNR time series is defined as [102, 103],

$$\rho(t) = \frac{z(t)}{\sigma}, \quad (1)$$

where $z(t)$ is the complex matched filter [102, 103], defined as,

$$z(t) = 2 \int_{-\infty}^{\infty} \frac{\tilde{s}(f)\tilde{h}^*(f)}{S_n(|f|)} e^{2\pi i f t} df. \quad (2)$$

Here, $\tilde{s}(f)$ is the Fourier transform of the detector strain data, $\tilde{h}(f)$ is the Fourier transform of the template waveform, and $S_n(f)$ is the power spectral density (PSD) of the strain data. In Eq. 1, σ is the normalization constant computed by the square root of the inner product of the template with itself.

Figure 1 compares the detector strain and the SNR time series produced by matched filtering for the first real detected BBH GW event, GW150914. Here, the template waveform has the median component mass parameters estimated for the compact binary merger from Ref. [3]. It can be seen that the signal power is much more concentrated in the SNR time series, making it a simpler task for a deep learning model. Note that lower binary masses can significantly increase the signal duration in the detector strain, while the power in the SNR time series would mostly remain within 0.1 s duration. The chirp mass is defined as

$$\mathcal{M}_c = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}}. \quad (3)$$

In this work, we generate the templates for matched filtering using the SEOBNRv4_ROM [104] waveform approximant with a low-frequency cutoff of 18 Hz. We used a subset of the template bank used by the online GstLAL pipeline in the O2 observation run [105], which was subsequently used by the SPIIR pipeline in the O3 observation run. Our subset comprises all templates where the detector frame primary and secondary masses are within the bounds of $4 - 200 M_\odot$, resulting in a bank of 19,365 templates. This space was chosen to target BBH signals with source frame masses of $5 - 100 M_\odot$, to cover the edges of the parameter space, and to account for the cosmological redshift of higher mass sources.

B. Training datasets

In this work, we generate data for training and testing the deep learning model by generating Gaussian noise

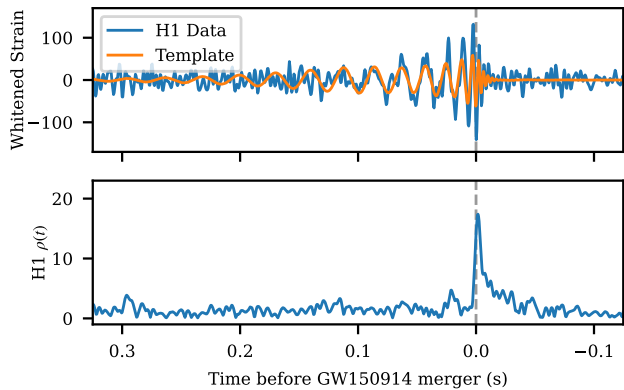


FIG. 1. Top panel: whitenened LIGO Hanford (H1) detector strain data containing GW150914 (blue), and the optimal waveform template that is generated using the SEOBNRv4_ROM approximant [104] and the median mass parameters from Ref. [3] (orange). The waveform template is non-spinning and has been whitenened and rescaled so that it visually overlaps with the signal in the strain data. Bottom panel: signal-to-noise ratio (SNR) time series for the event GW150914, from matched filtering using the strain data and waveform template shown in the top panel.

colored by the H1 and L1 LIGO detector PSDs. We also add simulated glitches that imitate real detector transient glitches to the noise background. For samples that contain a BBH signal injection, we project the signal onto the antenna pattern of each detector and inject them into the strain time series.

Our deep learning model makes predictions using the absolute value of the SNR time series from the H1 and L1 detectors as input. A PSD for each detector is calculated using the first seven days of strain data from the O3a observing run. The strain data used to generate these PSDs is sourced from the Gravitational Wave Open Science Center (GWOSC) [106]. Gaussian strain data for the training samples is generated from these PSDs using the PyCBC package [107] at a sampling frequency of 2048 Hz. To avoid the filter wraparound issue of a discrete matched filter [103], we generate noise samples with a length of 250 seconds, which is larger than the duration of the longest template waveform in our template bank, generated with the PyCBC package [107]. We select a 1-second region in the latter half of the 250-second sample as the second of data that encompasses the training sample, and here we inject any gravitational wave signal or simulated glitch injections. This 1-second region is the second that gets saved as the SNR time series training sample after matched filtering.

We simulate and inject a random glitch to the simulated detector strain at random positions in the noise. This work uses a set of glitch injections modeled by Gaussian, sine-Gaussian, and ring-down signals, as used in Refs. [76–78, 85] to simulate glitches in detector strain data. Some works use simulated glitches with additional

Parameter	Gaussian	Sine-Gaussian	Ring-down
h_0	$[10^{-21}, 5 \times 10^{-21}]$	$[10^{-21}, 5 \times 10^{-21}]$	$[10^{-22}, 10^{-21}]$
τ	$[0.0005, 0.005]$	-	-
f_0	-	$[40, 300]$	$[5, 20]$
Q	-	$[40, 300]$	$[12, 20]$

TABLE I. Parameter ranges for injected glitches. The amplitude scaling factor, h_0 , is sampled uniformly in base-10 log-space, and the other parameters are sampled uniformly in linear space.

signal morphologies [77, 78, 108]; however, the signals used in this paper are selected based on their similar morphology to BBH GWs to enhance our deep learning model’s detection ability. The Gaussian simulated glitches are defined by:

$$h(t) = h_0 \times \exp\left(-\frac{(t-t_0)^2}{2\tau^2}\right), \quad (4)$$

which has an amplitude scaling factor h_0 , a characteristic width τ , and a peak time t_0 . The sine-Gaussian glitches are represented as a sinusoidal signal with an amplitude that is modulated by a Gaussian,

$$h(t) = h_0 \times \sin(2\pi f_0(t-t_0)) \times \exp\left(-\frac{(t-t_0)^2}{2\tau^2}\right), \quad (5)$$

where f_0 is the central frequency, and the characteristic width is derived from the central frequency and quality factor, Q :

$$\tau = \frac{Q}{\sqrt{2\pi}f_0}. \quad (6)$$

Lastly, the ring-down glitch type is modeled as a damped sinusoid, with input parameters matching the sine-Gaussian signal,

$$h(t) = h_0 \times \sin(2\pi f_0(t-t_0)) \times \exp\left(-\frac{(t-t_0)}{2\tau}\right). \quad (7)$$

The bounds for each parameter used to generate these glitches can be found in Table I. All parameters are sampled uniformly within these bounds except for the amplitude scaling factor, which is sampled uniformly in base-10 log space. From large-scale tests, this parameter space results in a small set of glitches that can produce little to no effect on the SNR time series, up to peaks that reach an SNR of approximately 150.

The training dataset contains an equal number of samples containing a signal and samples that contain only noise. Additionally, in the training dataset, each detector has a 30% chance of injecting a glitch into each strain sample, where the type of glitch is randomly selected and of equal probability. This chance is independent for each detector, meaning there is a 9% chance of a random glitch occurring in both detectors for a given sample.

Parameter	Prior	Minimum	Maximum
Primary Mass, m_1 (M_\odot)	$m_1^{-2.35}$	5	100
Secondary Mass, m_2 (M_\odot)	m_2^1	5	m_1
Component Spins, S_z	Uniform	-0.998	0.998
Polarization Angle (rad)	Uniform	0	2π
Right Ascension (rad)	Uniform	0	2π
Declination (rad)	Cosine	$-\pi/2$	$\pi/2$
Inclination Angle (rad)	Sine	0	π
Luminosity Distance (Mpc)	Comoving	0	15000

TABLE II. Parameter ranges and sampling priors for simulated gravitational-wave injections. “Comoving” refers to a uniform in comoving volume prior.

The glitches are injected into the simulated noise before matched filtering. They are positioned with t_0 in the first 0.6 seconds of the 1-second time region that becomes the SNR time series input to the model. This accounts for the delay in the glitch response in the SNR time series due to the lengths of the templates.

We use an astrophysically distributed parameter space for the signal injections, similar to what was used in the offline analyses for GWTC-3 [6], and the parameter ranges and parameter priors for these injections can be seen in Table II. The component masses are sampled such that $m_1 > m_2$, each sampled from a power-law distribution with powers of -2.35 and 1, respectively. In Table II, the mass ranges are presented in the source frame, and the component masses are projected into the detector frame based on the sampled extrinsic parameters. Aligned spins are used and sampled uniformly.

Using astrophysical distributions when sampling these injections results in a significant number of injections with an SNR so low that the signal cannot be recovered with the current detectors. To account for this, we discard any injections where the estimated network SNR is less than 6. To calculate this estimated SNR and to inject the signals into the simulated strain, we use the SEOBNRv4_ROM waveform approximant [104] with a low-frequency cutoff of 18Hz. After generating the plus and cross polarizations of each injection, they are projected into the strain of the H1 and L1 detectors using the injection’s sky position and orientation and the detector antenna patterns.

Figure 2 shows the detector frame chirp mass distribution of this injection sampling for injections that pass the minimum network SNR threshold. There are noticeably fewer injections at high chirp masses, so we set a chirp mass limit for where we study our search method’s sensitivity, as our training dataset will not have enough samples to train the deep learning model sufficiently. This limit is indicated in Fig. 2 as a chirp mass of $110 M_\odot$, where only 5% of injections exist with a higher chirp mass.

We sample 100,000 unique signal injections, each injected into unique strain samples. The dataset consists of

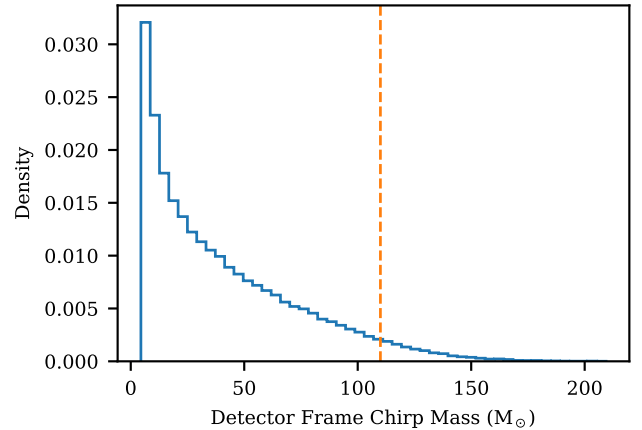


FIG. 2. Density histogram presenting the distribution of the detector frame chirp mass of signal injections in the training dataset after the minimum network SNR threshold is applied (blue). Only 5% of injections in the dataset have a chirp mass higher than the orange dotted line, which sits at a chirp mass of $110 M_\odot$. Our testing and analyses do not consider signals above this chirp mass, as our model has not been trained on enough samples in this region to expect good detection sensitivity.

10 different SNR time series produced by different templates for each signal injection. To select the ten templates for each signal, we sort the templates in the template bank by chirp mass and extract the adjacent 250 templates on either side of the chirp mass of the injection. We compute the overlap of each template with the simulated injection waveform, weighted by the PSD of the H1 detector, as it is the least sensitive detector. A template is discarded if the resulting overlap is below 0.5. From the remaining set of templates, we select the template with the highest overlap to the simulated injection, and we randomly sample the nine remaining templates from those that passed the overlap cut-off. This results in a dataset of 1,000,000 SNR time series samples that contain signal injections. Additionally, the training dataset includes 1,000,000 SNR time series samples without any signal injections, which includes 100,000 unique strain samples, and we randomly sample 10 templates from the template bank for each strain sample. Of the 2,000,000 samples in the training dataset, 20% are used for the validation dataset during training.

When generating each SNR time series of the samples containing an injection, the position of the merger time is randomly sampled to be in the central 0.9 seconds of the 1-second input window of the deep learning model. This is beneficial as it trains the deep learning model to search for patterns representing the existence of a GW signal across most of its input, which becomes crucial in constructing our ranking statistic in Sec. IID.

C. Deep learning model

The architecture of our deep learning model is designed to identify the complex features present in the SNR time series in the presence of noise, signal injections and glitches. We treat this problem as a pattern recognition problem, where we attempt to detect the presence of a signal separately in each detector and then determine if the signal is coincident between the other detectors.

From our training dataset, the input to our model is two SNR time series samples of dimension 1×2048 each, one sample for each of the H1 and L1 detectors. The model output is a single prediction value indicating whether the input sample contains a gravitational wave signal. The model output uses a sigmoid activation function to constrain the output to between 0 and 1, where ‘1’ represents a sample with a signal present.

A broad overview of our deep learning model can be seen in Fig. 3, which shows how we treat the detection problem in each detector separately using the detector blocks. The model concatenates the output of the two single-detector blocks and then uses a series of fully connected dense layers to assert if there is a coincident signal present. Additionally, we implement a ‘time-delay’ layer which takes the two detector inputs and computes the time between the SNR peak of each sample in units of light travel time, where the light travel time is the maximum time separation of a coincident gravitational wave signal based on the distance between the H1 and L1 detectors. This layer is implemented to give the model additional confidence when detecting gravitational wave signals where the time-delay value is expected to be ≤ 1 unless a glitch is present or the signal is weaker than a noise peak.

Each detector block has the same architecture, differing only by the weights and biases of each layer based on the data from each detector having separate PSDs, and this architecture can be seen in Fig. 4. This block of layers implements convolutional neural network (CNN) layers [109] to identify the patterns present in the data due to a gravitational wave signal. Combined with the CNN layers, we use a residual neural network (ResNet) structure [110], Fig. 5, which implements a skip connection to prevent overfitting in a complex model. At the start of the detector blocks, the CNN layers have 32 filters and a kernel size of 32. The CNN layers in the ResNet layers also have 32 filters but have a kernel size of 3, use padding to make the output shape equal to the input shape, and include the rectified linear unit (ReLU) activation function [111]. The maximum pooling layers in the detector blocks have a pool size of 8 and a stride of 2. Motivated by [48], we utilize Group Normalization layers [112] with a group size of 8 for normalizing across the channels, rather than the batch dimension, to aid in faster and more stable training. Lastly, we reduce the dimensionality of the output of the detector blocks using the flatten layer.

In the final stages of our model, it takes the outputs

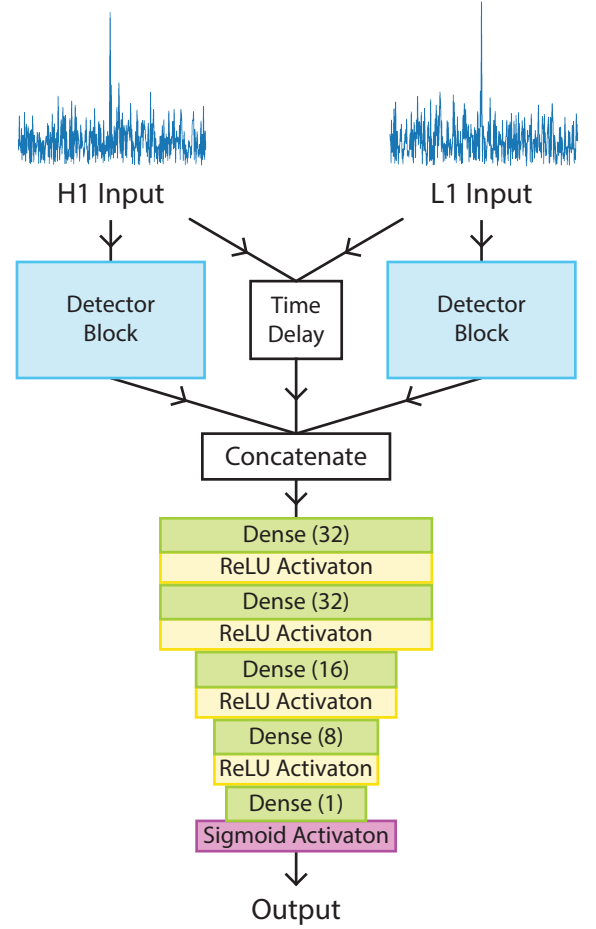


FIG. 3. Flow diagram of the overall model architecture. The model takes the SNR time series from the Hanford (H1) and Livingston (L1) detectors as inputs and feeds them into a detector block subarchitecture, which is depicted in Fig. 4. The number in brackets for each dense layer is the number of units, or neurons, in that layer.

from the time-delay layer and the two detector blocks and concatenates them before feeding them through a series of dense layers. The purpose of these layers is to take the condensed representation from the two detector blocks, consider the time-delay information, and assign significance to the model output based on its inputs. We use a concatenation layer so that both detector block outputs have their complete encoded representation present and so that this stage of the model can learn the coincidence, or lack thereof, between the two detectors.

Our deep learning model was built, trained, and tested using the TensorFlow [113] package. To train the model, we optimize by minimizing the binary cross-entropy loss function using the Adam optimizer [114] with an initial learning rate of 10^{-3} . The model training process reduces the learning rate by a factor of 10 if the loss does not decrease after 25 training epochs, down to a minimum learning rate of 10^{-6} , and the model training is exited after 200 epochs. This training process takes 7 hours on

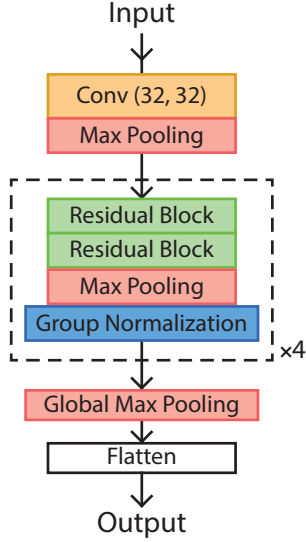


FIG. 4. Architecture of the detector blocks used in our deep learning model. The residual block subarchitecture can be found in Fig. 5. The dotted box indicates that the block of layers is repeated four times. However, the final repetition does not include the max pooling layer due to the global max pooling layer used before the output. The orange block indicates a convolutional layer, with the values in brackets indicating the layer’s number of filters and kernel size, respectively.

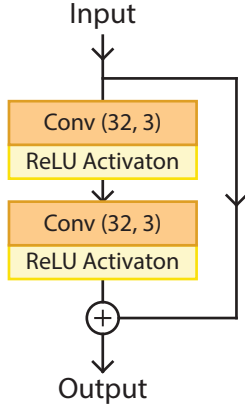


FIG. 5. Architecture of the residual neural network blocks used in the detector blocks of our deep learning model. The orange blocks indicate convolutional layers, with the values in brackets indicating the layer’s number of filters and kernel size, respectively.

a single NVIDIA A100 GPU.

Following training, we remove the sigmoid function from the model output and replace it with a linear function. This is motivated by some past work looking at removing bounds on the model’s output for gravitational wave detection [38, 39, 43, 48], and because the model’s float 32 precision results in predictions on many different inputs being rounded to 0 or 1 with the inclusion of the sigmoid function, having no significance between predic-

tions of the same value. Replacing the sigmoid function with the linear function allows the model output to have an unbounded output to address this issue.

D. Search pipeline

To go from our trained deep learning model to detecting gravitational wave events, we have set up a search method for producing triggers. Using this search method, we can define our ranking statistic and assign confidence to a trigger through a false alarm rate (FAR).

Our detection method uses a two-step process to produce a trigger on some data. The first step is to perform matched filtering on the strain time series from multiple detectors using every template in the template bank. This is followed by a coincident peak-finding search, which is aligned with how existing search pipelines produce triggers. The peak-finding search produces a network SNR, accounting for the light travel time between detectors for each template for every 1-second window in the data. At each second, we order the templates based on the network SNR. We also impose a minimum SNR of 4 in at least one detector so that we do not produce triggers when the SNR is too low for a detection to be made.

Since our model is the most accurate to templates with the highest signal response, we only make predictions with our model on the SNR time series from the ten highest network SNR templates for each 1-second window, based on the peak-finding outlined above. This also helps to manage the computational cost of this method, as the number of predictions will be orders of magnitude lower than if we made predictions over every template, which was found to be unnecessary. Since gravitational wave detector data can be considered as a continuous stream, we make a time series of 16 predictions per template instead of only one, as the peak identified can exist anywhere within the 1-second window, and our model will be less confident when peaks due to gravitational wave signals occur at the edges of its input window. These 16 predictions are performed at an inference rate of 16Hz, indicating they are overlapping predictions, and each one shifts across the SNR time series by 1/16th of a second. Additionally, the 8th of the 16 predictions is centred at the peak SNR in the H1 detector for the produced trigger. For each of the ten templates involved in creating a trigger, we calculate the average of the 16 predictions. This allows us to reject short transient spikes in the model predictions and promote longer-duration high outputs from the model indicative of a signal existing in the input window.

The set of averaged predictions for each of the ten templates is then clustered down to a single ranking statistic value from a single template. We select the template with the highest average prediction for our trigger, and this average prediction is used as our ranking statistic.

Using this method, we can perform analyses similar

to current search pipelines. This includes background runs where we characterize our search method and deep learning model, allowing us to assign an FAR to a trigger, and injection runs where we can determine the model's sensitivity to detecting gravitational wave signals.

III. TESTING AND RESULTS

In this section, we look at taking the search method that we have outlined in Sec. II and test its ability to have a stable ranking statistic, its sensitivity performance on a set of BBH injections, and how accurately it can produce SNR and chirp mass estimates of a BBH signal that it detects. To do this, we use a consistent simulated noise background and generate a set of astrophysical injections.

A. False alarm rate assignment

To properly analyze our search method's ability to detect BBH mergers, we first have to characterize our ranking statistic by collecting a background so that a false alarm rate can be assigned to triggers.

We prepare a dataset of 6×10^6 seconds, or ~ 70 days, of simulated noise background with no injected gravitational wave signals to characterize our ranking statistic. Since our model is trained on simulated noise sampled from the PSDs of H1 and L1 from the first week of the O3a observing run, this new noise background uses the PSDs of each detector from the second week of O3a. The simulated glitch rate for this noise background is 0.8 per minute for H1 and 2.67 per minute for L1. This glitch rate is 2.67 times the median for the O3a observing run [6]. This multiplier matches our glitch population with the glitch rate of a single-detector glitch with an SNR $\rho > 6.5$ in the O3 observing run.

We then use the search method outlined in IID to analyze this noise background and construct an estimated distribution of our ranking statistic on noise. Figure 6 presents a cumulative plot of our ranking statistic, which demonstrates how false alarm rates are assigned using the ranking statistic. Since our noise background consists of 6×10^6 seconds of data, we can only confidently assign false alarm rates to 1.67×10^{-7} Hz.

We implement a linear interpolation and extrapolation to our noise background to properly analyze our ability to detect gravitational wave signals, as shown in Fig. 6. This conservative extrapolation is calculated by fitting a line to the points on the cumulative background below an FAR of 10^{-5} Hz, and the fit has a coefficient of determination (R^2) of 0.982. We use the linear fit to assign an FAR when the ranking statistic is above ~ 11.45 , corresponding to an FAR of 10^{-6} Hz. We interpolate FARs from this threshold as the number of background data points is small, resulting in significant step changes in FARs as the ranking statistic increases.

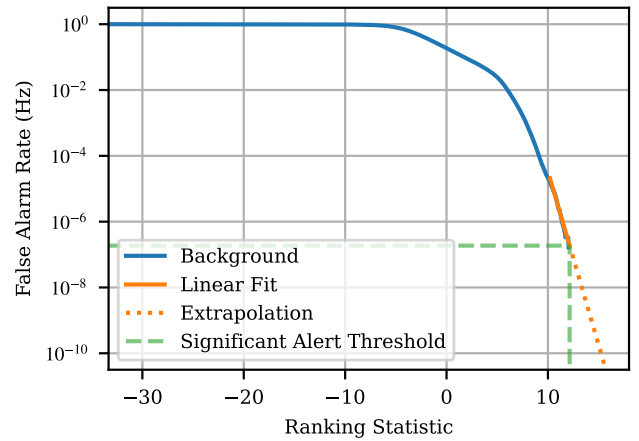


FIG. 6. Cumulative distribution of our ranking statistic (blue), plotted as a density to be able to assign false alarm rates. The solid orange line represents the linear fit on the tail of our background distribution, and the dotted orange line is the extrapolation of this fit used to assign lower false alarm rates. The green dotted line indicates the point at which the ranking statistic reaches a false alarm rate of 1 per 2 months based on the linear extrapolation.

Using the collected background for our ranking statistic and the method of assigning FARs to triggers, we can now test that it is robust to unseen noise. We do this by generating a new noise dataset using the PSDs from the third week of O3a to sample the detector strain and using the same glitch rate as the previous background. Figure 7 presents the cumulative counts of triggers for each inverse false alarm rate (IFAR). The triggers at each IFAR produced from this second analysis are within 3σ of the expected distribution.

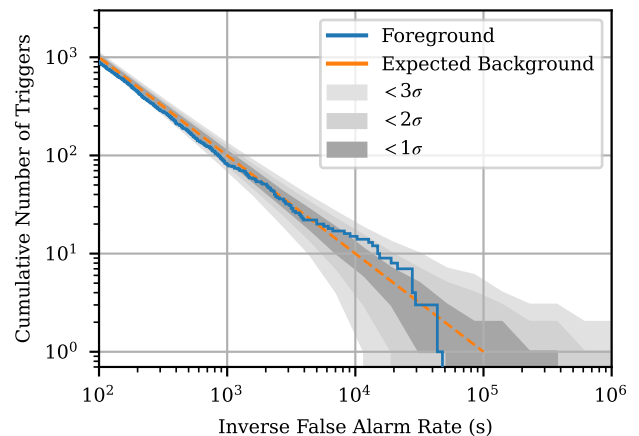


FIG. 7. Cumulative distribution of the inverse false alarm rate from triggers on Gaussian noise with simulated glitch injections. Our false alarm rate assignment defined on a collected background results in a distribution within 3σ of the expected distribution.

B. Injection results

In this section, we present the detection performance of our search method on a set of injected gravitational wave signals. Sec. III B 1 indicates the general sensitivity of our search method over the parameter space we search for. In section III B 2, we discuss the population of injections that we detect at an FAR of 1 per 2 months versus the population of injections that we fail to recover. Using these found injections, Sec. III B 3 presents how well our search method can recover the SNR of a population of signals. Lastly, Sec. III B 4 presents how our search method can make preliminary chirp mass estimates on each found injection based on the chirp mass of the template that was used to detect it.

1. Sensitivity

To analyze our search method's ability to detect simulated gravitational wave signals, we inject 10,000 BBH signals into a noise background. We then analyze the number of recovered signals at each FAR and the population of recovered signals at the significant alert threshold of 1 per 2 months. The injection set used in this testing matches the parameter distributions of the injection datasets used to analyze the sensitivity of the matched filtering search pipelines in Ref. [6]. This differs from our training dataset by incorporating isotropic spins and reducing the component mass lower limit to $2 M_{\odot}$. We use an upper spin magnitude limit of 0.998. When sampling injections, we incorporate a minimum estimated network SNR cutoff of 6, as in our training dataset, to remove any injections that are not expected to be recovered. We only perform tests on injections with both component masses $\geq 5 M_{\odot}$ to match our training parameter space. As mentioned in Sec. II B, only injections with a detector frame chirp mass $\leq 110 M_{\odot}$ are considered in our analysis due to the distribution of injections in our training dataset. Since these injections have isotropic spin components, we use the IMRPhenomPv2 [115] approximant for estimating the injection SNRs and SEOBNRv4PHMpseudoFourPN [116] for injecting the signals into the noise strain. The detector PSDs used for the noise background and for estimating each injection's SNR are the same week 3 PSDs used for the test that verified our FAR assignment, and we use the same glitch rate from this previous test.

For injection run tests, triggers are produced at one per second as outlined in Sec. II D. When injecting simulated gravitational wave signals, we randomly set the merger time so that it does not always occur at the same point of a 1-second window. Additionally, we generate new continuous noise strain samples for each injection so that signals do not overlap. When detecting an injected signal, we assign the trigger with the highest significance (lowest FAR) within ± 1 second of the injected merger time.

The sensitivity of our search method can be seen in the receiver operating characteristic (ROC) curve of Fig. 8. This plot focuses on our search method's ability to detect low network SNR signals. It compares this with a baseline search method that uses the maximum network SNR over the complete template bank as a ranking statistic. We plot the sensitivity of the two search methods over subsets of the total injection set, where the subset is all samples with an estimated network SNR in the range of ± 0.1 around a network SNR of 6, 8 and 10. We can see that due to the injection of simulated transient glitches, the sensitivity of the baseline matched filtering method drops off rapidly. In contrast, our search method demonstrates the deep learning model's ability to be robust in the presence of simulated transient glitches. It should be noted that when the baseline search curves drop rapidly, they follow the edge of the shaded region in Fig. 8, which means it matches the sensitivity of a random classifier.

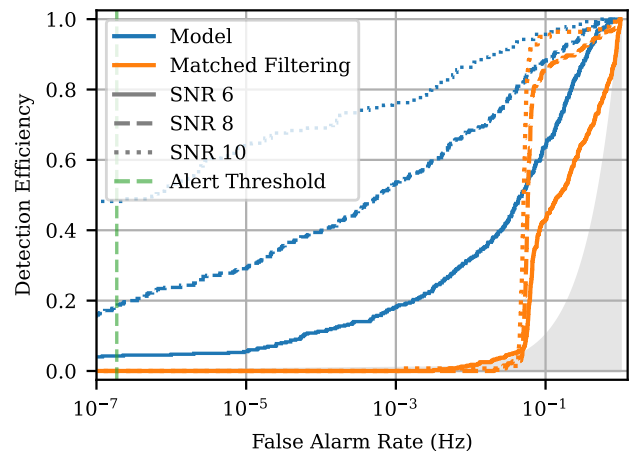


FIG. 8. Receiver operating characteristic curve indicating the relative sensitivity between our search method (blue) and one where the peak network SNR is used as a ranking statistic (orange). The injection samples used to calculate each curve are the injections from our test dataset within 0.1 of the estimated network SNRs in the legend. The shaded region indicates the performance of a random classifier on the test datasets, where the detection efficiency is equal to the false alarm rate.

We see that our search recovers 34% of all injections in a dataset that is heavily skewed toward injections with low network SNRs and low black hole masses. These initial sensitivity results are promising for applying this search method to detect BBH gravitational wave events, especially in the context of low network SNR signals. However, studies on real detector strain data and shared injection sets with existing search pipelines would be needed to confirm the feasibility of this search method.

Further optimizations could be explored for improving the search sensitivity in the future. These could include changing the training dataset parameter distributions, training loss and validation metrics, and new methods for the template selection of each sample in the training

dataset.

2. Missed and found injections

The performance of our search method on the injection tests can be seen in Fig. 9 in terms of missed and found injections against the Livingston detector’s effective distance and chirp mass. We classify injections as found if the trigger associated with the injection has an FAR less than 1 per 2 months. Our search method can detect 100% of BBH injections at an effective distance below 625 Mpc and more than 50% of signals when the distance is below 2.2 Gpc. Our results are also consistent with the expectation that heavier BBH mergers can be detected further away.

Notable in Fig. 9 is a small set of found injections at large effective distances in the Livingston detector relative to the general population of found injections. Intuitively, this would suggest that these injections have a low H1 effective distance and are being detected due to the higher SNR in the Hanford detector or because the injection occurs in a fortuitous noise realization where the Livingston SNR time series peaks above a low expected SNR value. Further analysis of these results indicates that some of these injections follow these expected explanations, however, there are still some outlier events in the relationship between effective distance and SNR. These remaining outliers are found to have high component spins, which results in spin-induced orbital precession. Because of this, the effective distance calculation that assumes a constant inclination is no longer valid, as the precession causes an inclination angle that varies with time. Further evidence of this can be seen in the GWTC-3 BBH injection set [117], which has the same priors as the injection dataset used in this work.

Figure 9 also shows three missed injections at low effective distances where all other surrounding injections are classified as found. The missed injections at a chirp mass of $33 M_{\odot}$ and $65 M_{\odot}$, and effective distance of 1100 Mpc and 1500 Mpc, respectively, are missed due to the presence of loud glitches in one detector, and a low estimated SNR in the other detector, resulting in no opportunity for the model to make a confident detection. The other missed injection at a chirp mass of $20 M_{\odot}$ and effective distance of 750 Mpc is missed by our search method as it is a single-detector event with an estimated SNR of 2 in H1 and 19.7 in L1, due to the source binary orientation and sky position. Not detecting this injection results from only considering a two-detector search method and not intentionally training our deep-learning model for this circumstance.

3. SNR recovery

Figure 10 demonstrates how accurately our search method recovers the SNR of a found injection in both

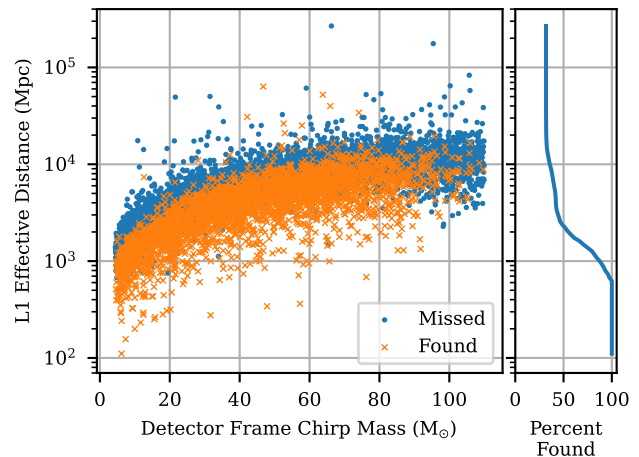


FIG. 9. Missed and found injections from our test dataset. Injections are classified as found if the false alarm rate is less than 1 per 2 months.

the Hanford and Livingston LIGO detectors. The recovered SNR comes from the template’s SNR time series, which produced the trigger. When sampling the signal injections, we compute the estimated SNRs in each detector by perfectly matching a projected injection waveform with itself using the same PSD as the noise background. Here, we have only included points where the H1 and L1 detectors are expected to recover an injection with a single-detector SNR greater than 4. The error, or fractional difference, of the recovered SNR is defined as

$$\text{error} = \frac{\text{recovered SNR} - \text{estimated SNR}}{\text{estimated SNR}}. \quad (8)$$

We see in Fig. 10 that our search method generally recovers the estimated SNR in each detector, which reinforces the selection of our template bank despite it being designed initially for a different observing run PSD than what our noise background is sampled from. Additionally, there is more spread at low SNRs for the LIGO Hanford detector, which is likely caused by the fact that it is less sensitive than the LIGO Livingston detector. This results in injections having a typically lower SNR in H1 than L1, and results in low network SNR injections with a low estimated SNR in L1 having an even lower SNR in H1 and thus not being able to be found by our search method at the 1 per 2-month threshold.

4. Chirp mass recovery

Since our search method incorporates templates with matched filtering, we can make preliminary estimates for the detector frame chirp mass of a source based on the chirp mass of the template used to produce the trigger. We present the recovered chirp mass against the expected

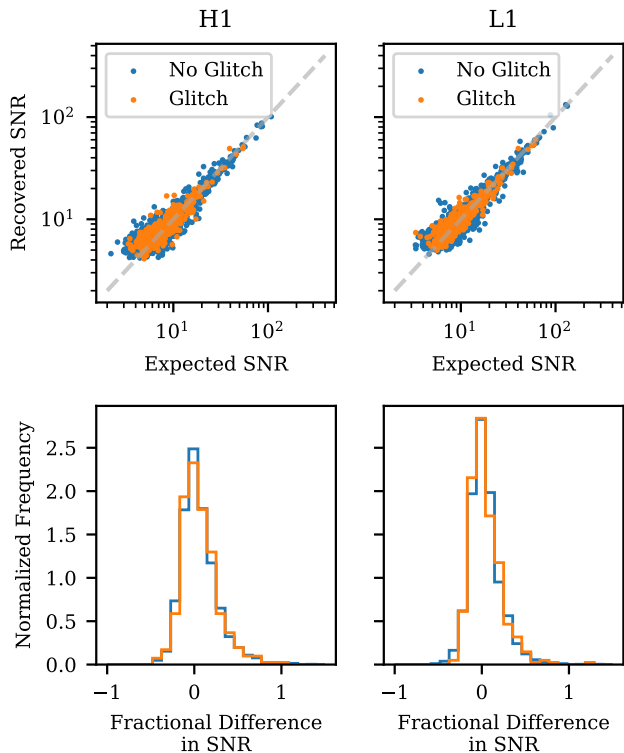


FIG. 10. Top panels: recovered and estimated SNRs for the found injections in our dataset in each interferometer: LIGO Hanford (H1) and LIGO Livingston (L1). The gray dashed lines indicate the diagonal where the recovered equals the estimated SNR. Bottom panels: fractional differences of the SNR in each detector, computed using Eq. 8. The orange points are injections where one of the detectors contained a simulated glitch within ± 1 second of the merger.

chirp mass for the found injections in Fig. 11. The fractional difference in the recovered chirp mass is computed in the same form as Eq. 8.

There is a notable spread in the recovered chirp mass for higher chirp mass injections, an expected feature of the BBH mass parameter space [13, 14]. This spread in recovered chirp mass for these higher-mass BBH signals is due to the low density of templates in this region of the template bank, as only a small number of templates are needed to search in this region sufficiently. This is due to these signals having a very short duration and a small frequency spectrum.

It is also clear from the fractional difference distributions in Fig. 11 that there is systematically no difference in the recovered chirp mass for injection times that include a glitch or not. This behavior can be due to several factors, including glitches existing in an adjacent trigger to the injection trigger, glitches having low strength, or glitches existing on the edge or outside the frequency spectrum of the template that produces the trigger.

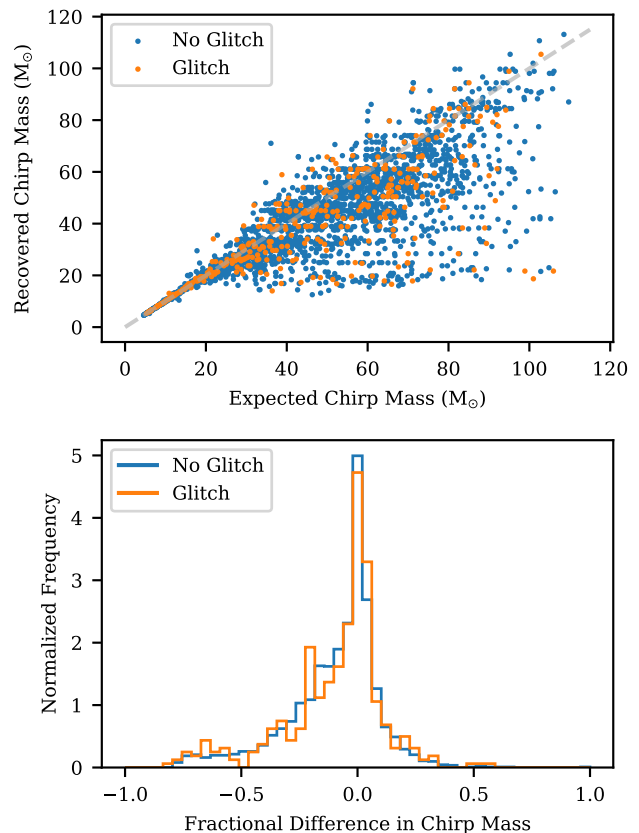


FIG. 11. Top panel: recovered vs expected detector frame chirp mass of the injections found by our search method. The gray dashed line indicates the diagonal where the recovered equals the expected chirp mass. Bottom panel: Fractional difference in the chirp mass computed in the same form as Eq. 8. The recovered chirp mass is retrieved from the template that produces the trigger associated with an injection. The orange points are injections where one of the detectors contained a simulated glitch within ± 1 second of the merger.

IV. CONCLUSION

In this paper, we have presented the results of a two-detector deep learning search method that aims to detect a broad parameter space of binary black hole gravitational wave signals from the SNR time series produced in matched filtering. We have demonstrated that our search method is robust against three representative short-duration glitches in gravitational wave strain. Additionally, we have demonstrated a promising level of detection sensitivity, motivating future work. Lastly, we have shown that by incorporating matched filtering into our search method, we can produce SNR and chirp mass estimates for detected gravitational wave mergers with comparable accuracy to existing search pipelines, despite using the output of our model to select triggers.

Comparing our sensitivity to the existing matched filtering pipelines is limited due to the pipeline search re-

sults involving real detector noise. However, we have demonstrated the potential robustness of our search methods to non-stationary detector data by incorporating simulated transient glitches into our noise.

Following these results, the next logical step would be to apply this search method to real detector noise for a direct comparison with existing search pipelines and to have an opportunity to identify real gravitational wave events. Extensions of this work would include detecting more gravitational source types, like BNS and NSBH mergers, where current deep-learning approaches have limitations. Lastly, implementing this search method in an online, real-time detection scenario should be considered, as matched filtering search pipelines already exist, and we could efficiently access the SNR time series from them.

ACKNOWLEDGMENTS

We wish to acknowledge the support of the Australian Research Council Centre of Excellence for Gravitational-Wave Discovery (OzGrav, Project No, CE170100004). This work was performed on the OzSTAR national facility at Swinburne University of Technology. The OzSTAR program receives funding in part from the Astronomy National Collaborative Research Infrastructure Strategy (NCRIS) allocation provided by the Australian Government, and from the Victorian Higher Education State Investment Fund (VHESIF) provided by the Victorian Government. This material is based upon work supported by NSF’s LIGO Laboratory which is a major facility fully funded by the National Science Foundation.

-
- [1] J. Aasi *et al.* (LIGO Scientific), Advanced LIGO, *Class. Quant. Grav.* **32**, 074001 (2015), [arXiv:1411.4547 \[gr-qc\]](#).
 - [2] B. P. Abbott *et al.* (LIGO Scientific, Virgo), Observation of Gravitational Waves from a Binary Black Hole Merger, *Phys. Rev. Lett.* **116**, 061102 (2016), [arXiv:1602.03837 \[gr-qc\]](#).
 - [3] B. P. Abbott *et al.* (LIGO Scientific, Virgo), 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), [arXiv:1811.12907 \[astro-ph.HE\]](#).
 - [4] R. Abbott *et al.* (LIGO Scientific, Virgo), GWTC-2: Compact Binary Coalescences Observed by LIGO and Virgo During the First Half of the Third Observing Run, *Phys. Rev. X* **11**, 021053 (2021), [arXiv:2010.14527 \[gr-qc\]](#).
 - [5] R. Abbott *et al.* (LIGO Scientific, VIRGO), GWTC-2.1: Deep Extended Catalog of Compact Binary Coalescences Observed by LIGO and Virgo During the First Half of the Third Observing Run, (2021), [arXiv:2108.01045 \[gr-qc\]](#).
 - [6] R. Abbott *et al.* (LIGO Scientific Collaboration, Virgo Collaboration, and KAGRA Collaboration), GWTC-3: Compact Binary Coalescences Observed by LIGO and Virgo during the Second Part of the Third Observing Run, *Phys. Rev. X* **13**, 041039 (2023).
 - [7] F. Acernese *et al.* (VIRGO), Advanced Virgo: a second-generation interferometric gravitational wave detector, *Class. Quant. Grav.* **32**, 024001 (2015), [arXiv:1408.3978 \[gr-qc\]](#).
 - [8] T. Akutsu *et al.* (KAGRA), Overview of KAGRA: Detector design and construction history, *PTEP* **2021**, 05A101 (2021), [arXiv:2005.05574 \[physics.ins-det\]](#).
 - [9] M. Saleem *et al.*, The science case for LIGO-India, *Class. Quant. Grav.* **39**, 025004 (2022), [arXiv:2105.01716 \[gr-qc\]](#).
 - [10] A. H. Nitz, T. Dal Canton, D. Davis, and S. Reyes, Rapid detection of gravitational waves from compact binary mergers with PyCBC Live, *Phys. Rev. D* **98**, 024050 (2018), [arXiv:1805.11174 \[gr-qc\]](#).
 - [11] T. Dal Canton *et al.*, Real-time Search for Compact Binary Mergers in Advanced LIGO and Virgo’s Third Observing Run Using PyCBC Live, *Astrophys. J.* **923**, 254 (2021), [arXiv:2008.07494 \[astro-ph.HE\]](#).
 - [12] K. Cannon *et al.*, GstLAL: A software framework for gravitational wave discovery, *SoftwareX* **14**, 100680 (2021).
 - [13] B. Ewing *et al.*, Performance of the low-latency GstLAL inspiral search towards LIGO, Virgo, and KAGRA’s fourth observing run, *Phys. Rev. D* **109**, 042008 (2024).
 - [14] Q. Chu *et al.*, SPIIR online coherent pipeline to search for gravitational waves from compact binary coalescences, *Phys. Rev. D* **105**, 024023 (2022), [arXiv:2011.06787 \[gr-qc\]](#).
 - [15] F. Aubin *et al.*, The MBTA pipeline for detecting compact binary coalescences in the third LIGO–Virgo observing run, *Class. Quant. Grav.* **38**, 095004 (2021), [arXiv:2012.11512 \[gr-qc\]](#).
 - [16] B. Allen, χ^2 time-frequency discriminator for gravitational wave detection, *Phys. Rev. D* **71**, 062001 (2005).
 - [17] B. P. Abbott *et al.* (LIGO Scientific, Virgo), Effects of data quality vetoes on a search for compact binary coalescences in Advanced LIGO’s first observing run, *Classical and Quantum Gravity* **35**, 065010 (2018).
 - [18] N. Christensen, P. Shawhan, G. González, and (for the LIGO Scientific Collaboration), Vetoes for inspiral triggers in LIGO data, *Classical and Quantum Gravity* **21**, S1747 (2004).
 - [19] LIGO Scientific Collaboration, *GraceDB — The Gravitational-Wave Candidate Event Database*.
 - [20] B. P. Abbott *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral, *Phys. Rev. Lett.* **119**, 161101 (2017).
 - [21] Y. LeCun, Y. Bengio, and G. Hinton, Deep Learning, *Nature* **521**, 436 (2015).
 - [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks, in *Advances in Neural Information Processing Systems*, Vol. 25, edited by F. Pereira, C. Burges, L. Bottou, and K. Weinberger (Curran Associates, Inc., 2012).
 - [23] K. Simonyan and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition

- (2015), [arXiv:1409.1556 \[cs.CV\]](#).
- [24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016) <http://www.deeplearningbook.org>.
 - [25] D. George and E. A. Huerta, Deep Neural Networks to Enable Real-time Multimessenger Astrophysics, *Phys. Rev. D* **97**, 044039 (2018), [arXiv:1701.00008 \[astro-ph.IM\]](#).
 - [26] 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), [arXiv:1711.03121 \[gr-qc\]](#).
 - [27] E. A. Huerta *et al.*, Accelerated, scalable and reproducible AI-driven gravitational wave detection, *Nature Astron.* **5**, 1062 (2021), [arXiv:2012.08545 \[gr-qc\]](#).
 - [28] H. Gabbard *et al.*, Matching matched filtering with deep networks for gravitational-wave astronomy, *Phys. Rev. Lett.* **120**, 141103 (2018), [arXiv:1712.06041 \[astro-ph.IM\]](#).
 - [29] X. Fan *et al.*, Applying deep neural networks to the detection and space parameter estimation of compact binary coalescence with a network of gravitational wave detectors, *Sci. China Phys. Mech. Astron.* **62**, 969512 (2019), [arXiv:1811.01380 \[astro-ph.IM\]](#).
 - [30] A. Rebei *et al.*, Fusing numerical relativity and deep learning to detect higher-order multipole waveforms from eccentric binary black hole mergers, *Phys. Rev. D* **100**, 044025 (2019), [arXiv:1807.09787 \[gr-qc\]](#).
 - [31] T. D. Gebhard *et al.*, Convolutional neural networks: a magic bullet for gravitational-wave detection?, *Phys. Rev. D* **100**, 063015 (2019), [arXiv:1904.08693 \[astro-ph.IM\]](#).
 - [32] G. R. Santos *et al.*, Gravitational wave signal recognition and ring-down time estimation via Artificial Neural Networks, *Expert Systems with Applications* **207**, 117931 (2022).
 - [33] R. Corizzo *et al.*, Scalable auto-encoders for gravitational waves detection from time series data, *Expert Systems with Applications* **151**, 113378 (2020).
 - [34] Y.-C. Lin and J.-H. P. Wu, Detection of gravitational waves using Bayesian neural networks, *Phys. Rev. D* **103**, 063034 (2021), [arXiv:2007.04176 \[astro-ph.IM\]](#).
 - [35] D. S. Deighan *et al.*, Genetic-algorithm-optimized neural networks for gravitational wave classification [10.1007/s00521-021-06024-4](https://doi.org/10.1007/s00521-021-06024-4) (2020), [arXiv:2010.04340 \[gr-qc\]](#).
 - [36] W. Wei *et al.*, Deep Learning Ensemble for Real-time Gravitational Wave Detection of Spinning Binary Black Hole Mergers, *Phys. Lett. B* **812**, 136029 (2021), [arXiv:2010.15845 \[gr-qc\]](#).
 - [37] H. Xia *et al.*, Improved deep learning techniques in gravitational-wave data analysis, *Phys. Rev. D* **103**, 024040 (2021), [arXiv:2011.04418 \[astro-ph.HE\]](#).
 - [38] M. B. Schäfer *et al.*, Training strategies for deep learning gravitational-wave searches, *Phys. Rev. D* **105**, 043002 (2022), [arXiv:2106.03741 \[astro-ph.IM\]](#).
 - [39] M. B. Schäfer and A. H. Nitz, From one to many: A deep learning coincident gravitational-wave search, *Phys. Rev. D* **105**, 043003 (2022), [arXiv:2108.10715 \[astro-ph.IM\]](#).
 - [40] C. Verma *et al.*, Employing deep learning for detection of gravitational waves from compact binary coalescences, *AIP Conf. Proc.* **2555**, 020010 (2022), [arXiv:2110.01883 \[gr-qc\]](#).
 - [41] C. Ma *et al.*, Ensemble of deep convolutional neural networks for real-time gravitational wave signal recognition, *Phys. Rev. D* **105**, 083013 (2022), [arXiv:2204.12058 \[astro-ph.IM\]](#).
 - [42] C. Verma *et al.*, Can Convolution Neural Networks Be Used for Detection of Gravitational Waves from Precessing Black Hole Systems?, (2022), [arXiv:2206.12673 \[gr-qc\]](#).
 - [43] M. Andrews *et al.*, DeepSNR: A deep learning foundation for offline gravitational wave detection, (2022), [arXiv:2207.04749 \[gr-qc\]](#).
 - [44] J. Yan *et al.*, Boosting the efficiency of parametric detection with hierarchical neural networks, *Phys. Rev. D* **106**, 063008 (2022), [arXiv:2207.11583 \[astro-ph.IM\]](#).
 - [45] F. P. Barone, D. Dell'Aquila, and M. Russo, A novel multi-layer modular approach for real-time fuzzy-identification of gravitational-wave signals, *Machine Learning: Science and Technology* **4**, 045054 (2023).
 - [46] M. B. Schäfer *et al.*, First machine learning gravitational-wave search mock data challenge, *Phys. Rev. D* **107**, 023021 (2023), [arXiv:2209.11146 \[astro-ph.IM\]](#).
 - [47] P. Nousi *et al.*, Deep residual networks for gravitational wave detection, *Phys. Rev. D* **108**, 024022 (2023), [arXiv:2211.01520 \[gr-qc\]](#).
 - [48] E. Marx *et al.*, A machine-learning pipeline for real-time detection of gravitational waves from compact binary coalescences, (2024), [arXiv:2403.18661 \[gr-qc\]](#).
 - [49] A. Trovato *et al.*, Neural network time-series classifiers for gravitational-wave searches in single-detector periods, *Classical and Quantum Gravity* **41**, 125003 (2024).
 - [50] O. Zelenka, B. Brügmann, and F. Ohme, Convolutional neural networks for signal detection in real ligo data, *Phys. Rev. D* **110**, 024024 (2024).
 - [51] T. S. Yamamoto, S. Kuroyanagi, and G.-C. Liu, Deep learning for intermittent gravitational wave signals, *Phys. Rev. D* **107**, 044032 (2023).
 - [52] M. Tian *et al.*, Physics-inspired spatiotemporal-graph AI ensemble for the detection of higher order wave mode signals of spinning binary black hole mergers, *Machine Learning: Science and Technology* **5**, 025056 (2024).
 - [53] P. Chaturvedi *et al.*, Inference-Optimized AI and High Performance Computing for Gravitational Wave Detection at Scale, *Frontiers in Artificial Intelligence* **5**, 10.3389/frai.2022.828672.
 - [54] S. Sasaoka *et al.*, Comparative study of 1D and 2D convolutional neural network models with attribution analysis for gravitational wave detection from compact binary coalescences, *Phys. Rev. D* **109**, 043011 (2024).
 - [55] M. D. Morales *et al.*, Deep learning for gravitational-wave data analysis: A resampling white-box approach, *Sensors* **21**, 3174 (2021), [arXiv:2009.04088 \[astro-ph.IM\]](#).
 - [56] K. Kim *et al.*, Identification of Lensed Gravitational Waves with Deep Learning, *Astrophys. J.* **915**, 119 (2021), [arXiv:2010.12093 \[gr-qc\]](#).
 - [57] J. D. Álvares *et al.*, Exploring gravitational-wave detection and parameter inference using deep learning methods, *Classical and Quantum Gravity* **38**, 155010 (2021).
 - [58] A. Menéndez-Vázquez *et al.*, Searches for compact binary coalescence events using neural networks in the LIGO/Virgo second observation period, *Phys. Rev. D* **103**, 062004 (2021), [arXiv:2012.10702 \[gr-qc\]](#).

- [59] S. Fan *et al.*, Improving Gravitational Wave Detection with 2D Convolutional Neural Networks, in *2020 25th International Conference on Pattern Recognition (ICPR)* (2021) pp. 7103–7110.
- [60] N. Lopac *et al.*, Detection of Non-Stationary GW Signals in High Noise From Cohen’s Class of Time–Frequency Representations Using Deep Learning, *IEEE Access* **10**, 2408 (2022).
- [61] A. Ravichandran *et al.*, Rapid Identification and Classification of Eccentric Gravitational Wave Inspirals with Machine Learning, (2023), [arXiv:2302.00666 \[gr-qc\]](#).
- [62] M. Andres-Carcasona *et al.*, Searches for mass-asymmetric compact binary coalescence events using neural networks in the LIGO/Virgo third observation period, *Phys. Rev. D* **107**, 082003 (2023), [arXiv:2212.02829 \[gr-qc\]](#).
- [63] S. Jadhav *et al.*, Improving significance of binary black hole mergers in Advanced LIGO data using deep learning: Confirmation of GW151216, *Phys. Rev. D* **104**, 064051 (2021), [arXiv:2010.08584 \[gr-qc\]](#).
- [64] S. Jadhav, M. Shrivastava, and S. Mitra, Towards a robust and reliable deep learning approach for detection of compact binary mergers in gravitational wave data, *Machine Learning: Science and Technology* **4**, 045028 (2023).
- [65] T. Fernandes *et al.*, Convolutional neural networks for the classification of glitches in gravitational-wave data streams, *Classical and Quantum Gravity* **40**, 195018 (2023).
- [66] Y.-X. Wang *et al.*, Rapid identification of time-frequency domain gravitational wave signals from binary black holes using deep learning, (2023), [arXiv:2305.19003 \[gr-qc\]](#).
- [67] C. Murali and D. Lumley, Detecting and denoising gravitational wave signals from binary black holes using deep learning, *Phys. Rev. D* **108**, 043024 (2023).
- [68] W. Alhassan, T. Bulik, and M. Suchenek, Detection of Einstein telescope gravitational wave signals from binary black holes using deep learning, *Monthly Notices of the Royal Astronomical Society* **519**, 3843 (2022).
- [69] T. Marianer, D. Poznanski, and J. X. Prochaska, A semisupervised machine learning search for never-seen gravitational-wave sources, *Monthly Notices of the Royal Astronomical Society* **500**, 5408 (2020).
- [70] A. E. Koloniari *et al.*, New Gravitational Wave Discoveries Enabled by Machine Learning (2024), [arXiv:2407.07820 \[gr-qc\]](#).
- [71] H. Wang *et al.*, Gravitational-wave signal recognition of LIGO data by deep learning, *Phys. Rev. D* **101**, 104003 (2020), [arXiv:1909.13442 \[astro-ph.IM\]](#).
- [72] M.-Q. Jiang, N. Yang, and J. Li, Identify real gravitational wave events in the LIGO-Virgo catalog GWTC-1 and GWTC-2 with convolutional neural network, *Front. Phys. (Beijing)* **17**, 54501 (2022).
- [73] C. Bresten and J.-H. Jung, Detection of gravitational waves using topological data analysis and convolutional neural network: An improved approach, (2019), [arXiv:1910.08245 \[astro-ph.IM\]](#).
- [74] S. Choudhary *et al.*, Deep learning network to distinguish binary black hole signals from short-duration noise transients, *Phys. Rev. D* **107**, 024030 (2023).
- [75] J. Powell *et al.*, Classification methods for noise transients in advanced gravitational-wave detectors II: performance tests on Advanced LIGO data, *Class. Quant. Grav.* **34**, 034002 (2017), [arXiv:1609.06262 \[astro-ph.IM\]](#).
- [76] J. Powell *et al.*, Classification methods for noise transients in advanced gravitational-wave detectors, *Class. Quant. Grav.* **32**, 215012 (2015), [arXiv:1505.01299 \[astro-ph.IM\]](#).
- [77] N. Mukund *et al.*, Transient Classification in LIGO data using Difference Boosting Neural Network, *Phys. Rev. D* **95**, 104059 (2017), [arXiv:1609.07259 \[astro-ph.IM\]](#).
- [78] M. Razzano and E. Cuoco, Image-based deep learning for classification of noise transients in gravitational wave detectors, *Class. Quant. Grav.* **35**, 095016 (2018), [arXiv:1803.09933 \[gr-qc\]](#).
- [79] E. Cuoco, M. Razzano, and A. Utina, Wavelet-Based Classification of Transient Signals for Gravitational Wave Detectors, in *2018 26th European Signal Processing Conference (EUSIPCO)* (2018) pp. 2648–2652.
- [80] S. Bahaadini *et al.*, Machine learning for Gravity Spy: Glitch classification and dataset, *Information Sciences* **444**, 172 (2018).
- [81] R. Biswas *et al.*, Application of machine learning algorithms to the study of noise artifacts in gravitational-wave data, *Phys. Rev. D* **88**, 062003 (2013), [arXiv:1303.6984 \[astro-ph.IM\]](#).
- [82] R. E. Colgan *et al.*, Efficient Gravitational-wave Glitch Identification from Environmental Data Through Machine Learning, *Phys. Rev. D* **101**, 102003 (2020), [arXiv:1911.11831 \[astro-ph.IM\]](#).
- [83] R. Essick *et al.*, iDQ: Statistical inference of non-gaussian noise with auxiliary degrees of freedom in gravitational-wave detectors, *Machine Learning: Science and Technology* **2**, 015004 (2020).
- [84] M. Cavaglia, K. Staats, and T. Gill, Finding the origin of noise transients in LIGO data with machine learning, *Commun. Comput. Phys.* **25**, 963 (2019), [arXiv:1812.05225 \[physics.data-an\]](#).
- [85] M. Llorens-Monteagudo *et al.*, Classification of gravitational-wave glitches via dictionary learning, *Class. Quant. Grav.* **36**, 075005 (2019), [arXiv:1811.03867 \[astro-ph.IM\]](#).
- [86] F. Morawski *et al.*, Anomaly detection in gravitational waves data using convolutional autoencoders, *Mach. Learn. Sci. Tech.* **2**, 045014 (2021), [arXiv:2103.07688 \[astro-ph.IM\]](#).
- [87] S. Bini *et al.*, An autoencoder neural network integrated into gravitational-wave burst searches to improve the rejection of noise transients, *Classical and Quantum Gravity* **40**, 135008 (2023).
- [88] C. Chatterjee *et al.*, Using Deep Learning to Localize Gravitational Wave Sources, *Phys. Rev. D* **100**, 103025 (2019), [arXiv:1909.06367 \[astro-ph.IM\]](#).
- [89] A. McLeod *et al.*, Rapid Mass Parameter Estimation of Binary Black Hole Coalescences Using Deep Learning, (2022), [arXiv:2201.11126 \[astro-ph.IM\]](#).
- [90] C. Chatterjee *et al.*, Rapid Localization of Gravitational Wave Sources from Compact Binary Coalescences Using Deep Learning, *The Astrophysical Journal* **959**, 42 (2023).
- [91] C. Chatterjee and L. Wen, Premerger Sky Localization of Gravitational Waves from Binary Neutron Star Mergers Using Deep Learning, *The Astrophysical Journal* **959**, 76 (2023).
- [92] M. Dax *et al.*, Real-Time Gravitational Wave Science with Neural Posterior Estimation, *Phys. Rev. Lett.* **127**,

- 241103 (2021), [arXiv:2106.12594 \[gr-qc\]](#).
- [93] Q. Tang, N. Yang, and J. Li, Deep learning for parameter estimation of supermassive binary black holes with simulated LISA data, *Chinese Journal of Physics* **88**, 301 (2024).
 - [94] O. G. Freitas *et al.*, Comparison of neural network architectures for feature extraction from binary black hole merger waveforms, *Machine Learning: Science and Technology* **5**, 015036 (2024).
 - [95] J. Langendorff *et al.*, Normalizing Flows as an Avenue to Studying Overlapping Gravitational Wave Signals, *Phys. Rev. Lett.* **130**, 171402 (2023).
 - [96] C. McIsaac and I. Harry, Using machine learning to autotune chi-squared tests for gravitational wave searches, *Phys. Rev. D* **105**, 104056 (2022).
 - [97] P. Joshi, R. Dhurkunde, S. Dhurandhar, and S. Bose, Optimal χ^2 discriminator against modeled noise transients in interferometric data in searches for binary black-hole mergers, *Phys. Rev. D* **103**, 044035 (2021).
 - [98] A. H. Nitz, Distinguishing short duration noise transients in LIGO data to improve the PyCBC search for gravitational waves from high mass binary black hole mergers, *Classical and Quantum Gravity* **35**, 035016 (2018).
 - [99] L. A. Wainstein, V. D. Zubakov, and A. A. Mullin, *Extraction of Signals from Noise* (Prentice-Hall, London, 1962).
 - [100] B. P. Abbott *et al.* (LIGO Scientific, Virgo), A guide to LIGO–Virgo detector noise and extraction of transient gravitational-wave signals, *Classical and Quantum Gravity* **37**, 055002 (2020).
 - [101] D. Davis *et al.*, LIGO detector characterization in the second and third observing runs, *Classical and Quantum Gravity* **38**, 135014 (2021).
 - [102] S. Hooper, *Low-latency detection of gravitational waves for electromagnetic follow-up*, Phd thesis, The University of Western Australia (2013).
 - [103] B. Allen *et al.*, FINDCHIRP: An Algorithm for detection of gravitational waves from inspiraling compact binaries, *Phys. Rev. D* **85**, 122006 (2012), [arXiv:gr-qc/0509116](#).
 - [104] A. Bohé *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), [arXiv:1611.03703 \[gr-qc\]](#).
 - [105] D. Mukherjee *et al.*, Template bank for spinning compact binary mergers in the second observation run of Advanced LIGO and the first observation run of Advanced Virgo, *Phys. Rev. D* **103**, 084047 (2021).
 - [106] R. Abbott *et al.* (KAGRA, VIRGO, LIGO Scientific), Open Data from the Third Observing Run of LIGO, Virgo, KAGRA, and GEO, *Astrophys. J. Suppl.* **267**, 29 (2023), [arXiv:2302.03676 \[gr-qc\]](#).
 - [107] A. Nitz *et al.*, *gwastro/pycbc: v2.0.5 release of PyCBC* (2022).
 - [108] V. Skliris, M. R. K. Norman, and P. J. Sutton, *Real-time detection of unmodelled gravitational-wave transients using convolutional neural networks* (2024), [arXiv:2009.14611 \[astro-ph.IM\]](#).
 - [109] Y. LeCun *et al.*, Backpropagation Applied to Handwritten Zip Code Recognition, *Neural Computation* **1**, 541 (1989).
 - [110] K. He *et al.*, Deep Residual Learning for Image Recognition [10.1109/CVPR.2016.90](#) (2015), [arXiv:1512.03385 \[cs.CV\]](#).
 - [111] V. Nair and G. E. Hinton, Rectified Linear Units Improve Restricted Boltzmann Machines, in *Proceedings of the 27th International Conference on Machine Learning*, ICML’10 (Omnipress, Madison, WI, USA, 2010) p. 807–814.
 - [112] Y. Wu and K. He, Group Normalization, in *Computer Vision – ECCV 2018*, edited by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss (Springer International Publishing, Cham, 2018) pp. 3–19.
 - [113] M. Abadi *et al.*, *TensorFlow: Large-scale machine learning on heterogeneous systems* (2015), software available from tensorflow.org.
 - [114] D. P. Kingma and J. Ba, Adam: A Method for Stochastic Optimization (2014) [arXiv:1412.6980 \[cs.LG\]](#).
 - [115] M. Hannam *et al.*, Simple Model of Complete Precessing Black-Hole-Binary Gravitational Waveforms, *Phys. Rev. Lett.* **113**, 151101 (2014).
 - [116] S. Ossokine *et al.*, Multipolar effective-one-body waveforms for precessing binary black holes: Construction and validation, *Phys. Rev. D* **102**, 044055 (2020).
 - [117] LIGO Scientific Collaboration, Virgo Collaboration, and KAGRA Collaboration, GWTC-3: Compact Binary Coalescences Observed by LIGO and Virgo During the Second Part of the Third Observing Run — O3 search sensitivity estimates, [10.5281/zenodo.7890437](#) (2023).