

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: <https://orca.cardiff.ac.uk/id/eprint/174275/>

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Skliris, Vasileios, Norman, Michael R. K. and Sutton, Patrick J. 2024. Toward real-time detection of unmodeled gravitational wave transients using convolutional neural networks. *Physical Review D (particles, fields, gravitation, and cosmology)* 110 (10) , 104034. [10.1103/physrevd.110.104034](https://doi.org/10.1103/physrevd.110.104034)

Publishers page: <https://doi.org/10.1103/physrevd.110.104034>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See <http://orca.cf.ac.uk/policies.html> for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.



Towards Real-Time Detection of Unmodelled Gravitational-Wave Transients Using Convolutional Neural Networks

Vasileios Skliris,^{*} Michael R. K. Norman,[†] and Patrick J. Sutton[‡]
*Gravity Exploration Institute, School of Physics and Astronomy,
Cardiff University, Cardiff, United Kingdom CF24 3AA*

(Dated: December 6, 2024)

Convolutional Neural Networks (CNNs) have demonstrated potential for the real-time analysis of data from gravitational-wave detector networks for the specific case of signals from coalescing compact-object binaries such as black-hole binaries. Unfortunately, CNNs presented to date have required a precise model of the target signal for training. Such CNNs are therefore not applicable to detecting generic gravitational-wave transients from unknown sources, and may be unreliable for anticipated sources such as core-collapse supernovae and long gamma-ray bursts, where unknown physics or computational limitations prevent the development of robust, accurate signal models. We demonstrate for the first time a CNN analysis pipeline with the ability to detect generic signals – those without a precise model – with sensitivity across a wide parameter space and with useful significance. Our CNN has a novel structure that uses not only the network strain data but also the Pearson cross-correlation between detectors to distinguish correlated gravitational-wave signals from uncorrelated noise transients. We demonstrate the efficacy of our CNN using data from the second LIGO-Virgo observing run. We show that it has sensitivity approaching that of the “gold-standard” unmodelled transient searches currently used by LIGO-Virgo, at extremely low (order of 1 second) latency and using only a fraction of the computing power required by existing searches, allowing our models the possibility of true real-time detection of gravitational-wave transients associated with gamma-ray bursts, core-collapse supernovae, and other relativistic astrophysical phenomena.

I. INTRODUCTION

Gravitational-wave (GW) astronomy is now an established field of observational science. To date, the LIGO [1] and VIRGO [2] collaborations have published the details of approximately 90 detection candidates [3–9] over their first three observing runs. The detected signals originate from the binary inspiral and merger of two black holes [10], two neutron stars [11], or one object of each type [12].

Low-latency detection of candidate signals offers arguably the greatest potential scientific payoff, as the GW observations can trigger followup observations in other channels; *i.e.*, multi-messenger astronomy. For example, combined GW and electromagnetic observations of GW170817 - GRB 170817A [13] have yielded novel insights into the origin of heavy elements [14], neutron-star structure [15], GRB astrophysics and host environments [16], and the Hubble constant [17]. Electromagnetic followup of gravitational-wave signals requires very low latency analysis of the GW data - preferably at the second scale to capture the highest energy emissions (e.g., the prompt gamma and x-ray emission of GRBs). Current low-latency GW analysis techniques rely on hundreds of dedicated CPUs to achieve latencies of tens of seconds to minutes for automated alerts [18].

Recent work by a number of authors [19–22] has shown that a fundamentally different approach using CNNs has

the potential to analyse detector data for GW signals in real time (~ 1 s latency) using a single dedicated GPU. However, most CNNs demonstrated to date require a specific signal model for training (*e.g.* an analytic signal model for binary mergers [19–32], or catalogs of numerically computed signals for core-collapse supernovae [33–36]), and are therefore only capable of detecting signals matching that model. Many potential sources are governed by physics which is either unknown (*e.g.* the neutron star equation of state [37, 38]) and/or computationally intractable (*e.g.* the modelling of accretion-disk instabilities [39–41]); their transient signals are commonly known as gravitational wave bursts (GWBs). While the unknown physics governing GWBs makes the study of such signals exciting, it also poses a challenge: To fully explore the new GW window we need to be able to detect signals from the widest possible variety of sources without relying on precise models for training.

We address this challenge by proposing a novel CNN architecture that analyses not only the detector strain data directly but also the cross-correlation timeseries between detectors. By training the CNN with ‘featureless’ randomised signals, we are able to construct a neural network that detects coherence (amplitude and phase consistency) between detectors rather than specific signal shapes in individual detectors. We test our resulting analysis pipeline, which we name MLY (“Emily”), using real data from the LIGO-Virgo network and show that it is capable of detecting a variety of simulated GWB signal morphologies without being specifically trained for them, at sensitivities close to that of standard GWB searches, but at much lower latency and a tiny fraction of the computational cost.

^{*} SklirisV@cardiff.ac.uk

[†] NormanM5@cardiff.ac.uk

[‡] SuttonPJ1@cardiff.ac.uk

This paper is organised as follows. In Section II we give a brief review of applications of machine learning in gravitational-wave astronomy. In section III we present the architecture of MLY’s CNNs and describe the analysis procedure and training data. In Section IV we discuss how we optimize the training to maximize performance. In Section V we present the performance of MLY on both simulated and real LIGO-Virgo data. We discuss the implications of these results and next steps in Section VI.

II. MACHINE LEARNING IN GRAVITATIONAL-WAVE ASTRONOMY

The fields of gravitational-wave astronomy and deep learning have both advanced significantly in recent years. As such, there has been a confluence of research into their combination and a considerable body of work has developed. Artificial neural networks, including CNNs, autoencoders [42], generative adversarial networks [43], recurrent neural networks [44], attention-based networks like transformers [45], and various other architectures, have been applied to a variety of problems within gravitational-wave astronomy [46]. In this section we summarise a few of these efforts most closely related to the present work; see for example [25, 47, 48] for wider reviews.

Deep-learning studies in gravitational-wave astronomy have most commonly focused on signals from the inspiral, merger, and ringdown of binaries consisting of black holes and/or neutron stars. A number of groups have demonstrated such methods for the detection of binary merger signals [19–32, 49–55], as well as for parameter estimation - determining the source properties from a detected signal [23, 27, 28, 56–59]. Deep learning has also been used for fast binary waveform generation [60], which could greatly reduce the time required by traditional Bayesian parameter estimation techniques, and for denoising data around signals [42, 61].

In each of these deep-learning studies, the training of the network has relied on the availability of highly accurate models for gravitational wave signals from binary mergers [62]. The application of deep learning methods to more general types of gravitational-wave signals has been more limited, with core-collapse supernovae being the most prominent example. These studies have made use of either catalogs of gravitational-wave signals from numerical simulations of core collapse or phenomenological waveform models fit to such catalogs. For example, Chan *et al.* [33] trained a CNN using simulated gravitational-wave timeseries with core collapse supernovae signals drawn from a range of published catalogs covering both magnetorotational-driven and neutrino-driven supernovae, and measured the ability to both detect the signal and correctly classify the type. Iess *et al.* [34] considered the problem of distinguishing true signals from noise fluctuations (“glitches”) that are common in real detectors. They used signals drawn from nu-

merical catalogs combined with a simple phenomenological models for two glitch types to train CNNs to distinguish supernova signals from noise glitches.

Cavaglia *et al.* [63] uses supernova simulated morphologies to differentiate them from glitches in individual detectors. This work show a clever application of decreasing the false alarm rate when multiple detectors are observing.

Lopez *et al.* [36] (building on [35]) used a phenomenological model mimicking gravitational-wave signals from non-rotating core-collapse supernovae to train a complex mini-inception resnet neural network [64] to detect supernova signals in time-frequency images of LIGO-Virgo data.

Sasaoka *et al.* [65] [66] use gradient-weighted feature maps to train CNNs to recognise supernovae spectrograms. They utilised core-collapse supernovae waveforms from a number of catalogues.

Moreno *et al.* [67] use recurrent autoencoders for anomaly detection. Autoencoders attempt to learn a function to project elements drawn from an input distribution into a dimensionally reduced latent space and then reconstruct the original input element from this reduced latent space. Since the encoder and decoder are trained on a specific distribution, if they are fed an element from outside the distribution, there will be a larger difference between model input and output, indicating an anomaly. In a later work the same group [68] has developed this work into a potential offline search (GWAK) for anomalous detection.

We note that all of these examples, both for binary mergers and for supernovae, rely on having a signal model to train the deep network. As a consequence, their applicability is restricted to signals that are similar to the training data. While not an issue for binary mergers, this may be very important for supernovae where the simulations used for training rely on uncertain physics and numerical approximations and simplifications [69, 70]. And they are clearly not applicable to the more general problem of detecting gravitational-wave transients from as-yet unknown sources.

Shortly after the release of an early version of this work [71], Marianer *et al.* [72] presented a deep-learning algorithm that avoids relying on a signal model by instead using outlier detection. The authors trained a mini-inception resnet network [64] on the Gravity Spy data set [73, 74], which contains spectrograms of known noise glitches classified into categories. They then applied the CNN to spectrograms of LIGO data and used two methods of outlier detection to identify possible signals. This search was applied to a subset of public LIGO data from the first two observing runs; no signal candidates were found. To our knowledge this is the only other case to date of a deep-learning method capable of searching for generic gravitational-wave transients.

In a special category Szczepańczyk *et al.* [75] use CWb search as a base and independently enhance the search using a machine learning model. Although this is signif-

icantly different from our approach which uses machine learning as the base of the search.

In this paper we present a deep-learning technique that is capable of detecting generic transient gravitational-wave signals. Our approach differs from previous approaches in a key way: rather than training a CNN to recognise specific signal morphologies in the data streams, we construct CNNs that are designed to recognise *coherence* in amplitude and phase between two or more data streams. We then train the CNNs using simulated signals and noise glitches that both consist of *random* timeseries with properties drawn from the same distributions. Using the same waveform distributions to simulate both the signals and glitches prevents the CNNs from using the signal morphology to the classify input. Instead, the CNNs are forced to learn to measure consistency between detectors.

In the next section we describe the architecture of MLY’s CNNs and the training procedure. We then evaluate MLY by analysing data from the second LIGO-Virgo observing run. We will see that our trained pipeline has a detection efficiency approaching that of the standard LIGO-Virgo pipeline for detecting unmodelled gravitational-wave transients [76], but with higher speed and lower computational cost.

III. A CNN FOR UNMODELLED BURST DETECTION

Our goal is to be able to detect sub-second-duration GWBs in data from the three detectors of the LIGO-Virgo network, without prior knowledge of the signal morphology. A significant challenge is to distinguish real signals from the background noise transients, “glitches”, that are common in these detectors [77–79]. Typical GWB detection algorithms [80–83] do this by requiring candidate signals to be seen simultaneously in multiple detectors (simultaneously up to the light travel time between the detectors) and to be correlated between detectors. We follow this logic in our analysis by using a network architecture for MLY that combines the outputs of two different CNNs: one that detects coincident signals in multiple detectors (Coincidence Model - Model 1), and a second that detects correlation in phase and amplitude between the detectors (Coherence Model - Model 2). Model 1 takes as input the band-passed, whitened timeseries data from each detector. Model 2 takes the same band-passed whitened timeseries data as well as the Pearson correlation between each pair of band-passed whitened timeseries data [equation (1)]. Each model outputs a score on $[0, 1]$, where values near 1 indicate a signal and values near 0 indicate noise. The scores from the two models are multiplied together to give a combined score on $[0, 1]$.

In the following subsections, we first review the analysis procedure and the choice of training data. We then detail how the architecture for each model is selected.

A. Analysis Procedure

We analyse data from all three of the detectors in the LIGO-Virgo network: LIGO-Hanford (H), LIGO-Livingston (L), and Virgo (V). In our analysis we use two types of background noise. For training we use simulated Gaussian noise that follows the design curves for the LIGO and Virgo detectors [84]; the motivation for using simulated noise for training is explained in Section III B. For testing we use real LIGO-Virgo data publicly available from the GW Open Science Center (GWOSC) [85].

The LIGO-Virgo data from GWOSC are sampled at 4096 Hz. We downsample to 1024 Hz, allowing us to detect signals up to 512 Hz; this covers the most sensitive frequency range of the detectors and is sufficient for the purpose of demonstrating our CNN. (Since the trained network can process data much faster than real time, we could extend the analysis to higher sample rates. We leave this to future work.) We chose to focus on signal durations < 1 s by analysing data in 1 s segments. This covers many plausible signal models, including for example core collapse supernovae [86], perturbed neutron stars and black holes [12], and cosmic string cusps [87]. We could extend to longer durations as well, with a corresponding increase in latency.

The power spectral density (PSD) $S_\alpha(f)$ for each detector α is computed using Welch’s method, and used to whiten the corresponding data stream. We use 16 seconds to calculate the PSD, whiten and then keep the central second. Each data stream is also high-pass filtered at 20 Hz, giving a search band of $[20, 512]$ Hz. The Pearson correlation of each pair of detectors is given by

$$r_{\alpha\beta}[n] = \frac{\sum_{i=1}^N (d_\alpha[i] - \bar{d}_\alpha)(d_\beta[i+n] - \bar{d}_\beta)}{\sqrt{\sum_{j=1}^N (d_\alpha[j] - \bar{d}_\alpha)^2 \sum_{k=1}^N (d_\beta[k] - \bar{d}_\beta)^2}}. \quad (1)$$

Here $d_\alpha[i]$ is the band-passed whitened data timeseries for detector α , \bar{d}_α is the mean over N samples, and n is an integer time delay between detectors. The correlation is computed for all n corresponding to time delays of up to ± 30 ms with respect to the first detector, which is slightly larger than the maximum possible arrival time difference (± 27.3 ms) a GW signal can have in the three-detector LIGO-Hanford, LIGO-Livingston, and Virgo network.

The band-passed whitened data series and the correlation series are then fed into the two models. The scores from the two models are multiplied together to give a combined score on $[0, 1]$. In this way a candidate signal needs to score highly for both models; *i.e.*, showing both coincidence in multiple detectors and correlation between the detectors. In practice we find the scores of both models tend to be strongly peaked around 0 for noise and weak astrophysical signals, and strongly peaked around 1 for strong astrophysical signals.

To estimate the distribution of scores of the background noise, we repeat the analysis many times after time-shifting the data between detectors by an integer number of seconds. Since the time shift is much larger

than the longest possible time-of-light delay between detectors, it prevents a real GW signal from appearing in coincidence between multiple detectors. All coincident events in the time-shifted series can therefore be assumed to be uncorrelated and treated as background noise. This is a standard procedure in GW analysis; see *e.g.* [77].

To estimate the sensitivity to GWBs, we repeat the analysis after adding simulated signals to the data. In General Relativity, a GW has two polarisations, denoted $h_+(t)$ and $h_\times(t)$. The received signal $h_\alpha(t)$ of a given detector α is the combination

$$h_\alpha(t) = F_\alpha^+ h_+(t) + F_\alpha^\times h_\times(t) \quad (2)$$

where the antenna response functions $F_\alpha^{+,\times}$ are determined by the position and orientation of the source relative to the detector. We characterise the strength of the received signal by its network signal-to-noise ratio

$$\rho = \sqrt{\sum_\alpha 4 \int_0^\infty \frac{|\tilde{h}_\alpha(f)|^2}{S_\alpha(f)} df} \quad (3)$$

Generating simulated signals distributed isotropically over the sky and rescaling to different ρ values allows us to measure the distribution of CNN scores as a function of the signal-to-noise ratio of the signal.

B. Training Data

The choice of data used to train a CNN is often a critical factor for the CNN’s performance.

For the signal population we use white-noise bursts (WNBs) [77, 82]; these are signals where the h_+ and h_\times polarisations are independent timeseries of Gaussian noise that is white over a specified frequency range, multiplied by a sigmoid envelope. We select these as our training sample as they are effectively featureless. The bandwidth of each simulated (or “injected”) signal is selected randomly and uniformly over the range [40, 480] Hz. The duration of each injection is selected randomly and uniformly over the range [0.05, 0.9] s. Given their duration their central time is chosen randomly inside this 1-second interval in a way that they aren’t cropped. The injections are distributed uniformly over the sky and projected onto the detectors using equation (2). Finally, the signal is rescaled to a desired network signal-to-noise ratio ρ as defined in equation (3). In the rest of this section we will discuss different aspects of our training data and methods, that eventually gave us the current best model.

For the background population we find in practice that the best CNN performance is obtained by training with *simulated* detector noise and glitch-like instances, rather than real glitching detector noise. The simulated glitch-like instances are WNBs with parameters drawn from the same distribution as for GWBs, but independently between detectors (*i.e.*, a different WNB is used for each

detector). Using simulated background allows us to control the glitch rate in the training set; as we will show in Section IV C that we can vary this rate to improve the performance at low false alarm rates. Furthermore, using WNBs for the glitches prevents the CNNs from learning to distinguish GWBs from glitches based on the morphology; this is critical since we do not know the true morphology of GWBs.

In the remainder of this section we detail the training process for each model. All training data were generated using the MLY package [88] and its generator function, with elements of the PYCBC [89] and GWPY[90] package to project the signal onto the detectors and apply time-of-flight differences to the signals arriving at the various detector locations. For training the models we used KERAS [91]. More details and the codes are available at our repository [92] and the mly package [88].

1. Data types

We have two CNN models to train: the Coincidence Model (model 1) and the Coherence Model (model 2). Each model is trained with a dataset chosen to produce the best performance of the model for its assigned task.

All training samples consist of stationary background noise and optionally an injection into one of more of the detectors. The stationary background is Gaussian noise with power spectra that follow the design curves of the LIGO and Virgo detectors [84]. For the non-stationary glitch component we use the same WNB waveform family as for the GWB signals but either do the injection into a single randomly chosen detector or we inject into all three detectors but select the WNB parameters independently for each detector. We use a total of four types of samples:

- **Type 0:** Gaussian LIGO-Virgo noise, with no injections. Labeled as noise.
- **Type 1:** Gaussian LIGO-Virgo noise with coherent WNB injections in all detectors, simulating a real GWB. Labeled as signal.
- **Type 2:** Gaussian LIGO-Virgo noise with different (incoherent) WNB injections in each detector, simulating unrelated glitches or excess noise in each detector. The injections may be simultaneous or offset in time from each other to simulate simultaneous or nearly simultaneous glitches. Labeled as noise.
- **Type 3:** Gaussian LIGO-Virgo noise with a single WNB injection in a randomly chosen detector, simulating a glitch in a single detector. Labeled as noise.

The Coincidence Model (model 1) is trained using Type 0 (stationary background), Type 1 (GWBs), and Type 3 (single-detector glitches). The GWBs are injected

with network SNR values in the range [12,30]. The limits of this range are chosen based on the fact that lower SNR values in training increase the false alarm rates dramatically, while training SNRs higher than 30 are not required; we show later that the detection efficiency remains high for SNR values larger than those in the training set. Single-detector glitches (Type 3) have SNR values over the range [6,70].

The Coherence Model (model 2) is trained using Type 0 (stationary background), Type 1 (GWBs), and Type 2 (incoherent multi-detector glitches). Experimentation showed that training with GWB network SNRs in the range [10,50] and glitch network SNRs in the range [10,70] gives the best performance.

The Type 2 incoherent signals represent the extreme cases where glitches are present in the same 1-second interval in all three detectors. Real glitches occur independently in different detectors and need not be simultaneous to within the light travel time between detectors. To train our model to handle this case we define a quantity called “disposition” which is the range of central times of the glitches. For example, in the case of three detectors and a disposition T the glitches will be centred at times $T_0 - T/2$, T_0 , and $T_0 + T/2$ in the three detectors, with the order selected randomly. The central time T_0 is positioned randomly in the 1 second interval so that no signal gets cropped, and the injection durations are also restricted so that no signal gets cropped. We generate datasets for dispositions distributed uniformly over three different ranges: Type 2a has range of [0.1,0.5] seconds, Type 2b has range [0,0.1] seconds, and Type 2c has zero disposition to challenge the Coherence Model with coincident incoherent signals.

C. Network Architecture

1. Coincidence Model - Model 1

The first model is a single-input single-output residual neural network whose goal is to identify coincident signals that appear in at least two detectors. This model takes as input the whitened timeseries data from each of the three LIGO-Virgo detectors. The output is a score on [0, 1], where high values indicate signal and low values no signal.

Residual neural networks are proven to boost the performance of simpler CNNs by reducing the effect of vanishing gradients; the latter make deep CNNs lose contributions from their first layers and cause their efficiency to saturate. We adapt a network from [93] that compares different methods of using machine learning on time-series data and we optimise it using a genetic algorithm (see below). We find that the resulting residual neural networks outperform “ordinary” deep CNNs in our case. More specifically we optimise a simple CNN model with a relatively good performance (overall accuracy >95%) and then use it as a “residual block”, where

we feed the output of the block to its input. To find the hyper-parameters of the model we use a genetic algorithm [94] that trains many generations of different randomly initialised models and find which hyper-parameters increase the performance. The two main differences from the original model of [93] are a larger kernel size and the reduction in our residual blocks from three layers to two. We find that three residual blocks are optimal, as in [93]. Varying the number of filters has no obvious benefit so we retain the original number to make the model less computationally expensive.

In training we use early stopping, which stops the training when the loss does not improve for 20 consecutive epochs. We find that overall accuracy is further improved by using a cyclical learning rate [95] with the addition that the learning rate is halved when the loss does not decrease for 5 epochs. Finally, we save the model for every epoch in which the accuracy increases, so that the final model saved is the best one encountered over the training.

The final model is shown in Figure 1. It has three residual blocks, the first with 64 filters and the others with 128. At the end of each residual block we add the output of the first layer to the block output. By doing this the gradient can skip the intermediate layer reducing the vanishing effect. Since we use convolution, all layers are zero-padded to maintain the same size with the input and make this addition feasible. At the end of each layer we apply ReLU (rectified linear unit) activation followed by batch-normalisation. After the last residual block we flatten using global average pooling. We then use two dense (fully connected) layers with batch-normalisation before passing to the output layer, which uses softmax activation. We use binary cross entropy to calculate the loss.

2. Coherency Model - Model 2

The second model has two inputs and one output. The first input is the same whitened timeseries data fed to the first model, while the second input is the Pearson correlation of each pair of detectors, equation (1). This model has a binary classification output that is trained to return a measure of coherency among detectors on [0,1].

Measurements of correlation between detectors have long been a key ingredient of GWB detection pipelines; see *e.g.* [76, 82, 96–98]. We have explored training networks to infer correlation information from the strain data; however, this is not a natural operation for networks constructed from convolutional filters. Since the Pearson correlation is simple to compute and easy to digest by a feature-detecting algorithm, we choose to feed the correlation as a input to the model.

The strain and correlation inputs have their own separate branches which are eventually merged as shown in Figure 2. As for the first model, we use a genetic algorithm to optimise the hyper-parameters. Due to the

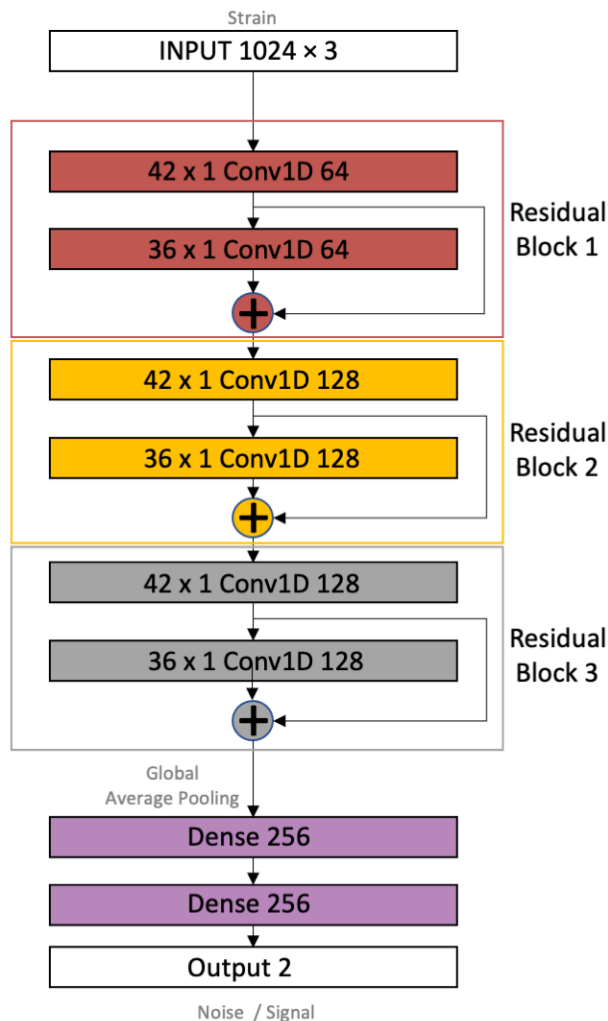


Figure 1. Coincidence model architecture (Model 1). The input consists of three 1024-sample whitened data channels from the H, L, and V detectors. This is passed through three consecutive two-layer residual blocks, flattened, then passed through two final dense layers. $M \times N$ indicates the size of the filters in each convolutional layer and 64/128/256 is the number of filters (convolutional layers) or nodes (dense layers).

small size of the correlation input, we find that its branch does not need to be as deep as the coincidence model, nor does the strain branch. We find that the performance is sensitive to the number of filters: increasing or decreasing the number in any of the convolutional layers can prevent the model from training. By contrast the kernel size has no significant effect on performance, except for some variation of the stability of training in some cases. The choice of kernel size was made based on how often those numbers appeared in the gene pool of the last generation of successful models in the genetic algorithm.

We use the same early stopping and cyclical learning rate choices as used for model 1.

The final model is shown in Figure 2. We pass the

strain input through three convolutional layers and the correlation data through two convolutional layers in a separate branch. (We also explored residual neural networks for the strain branch of the model but these gave no improvement in performance.) We flatten the outputs by global average pooling, which increases the performance slightly, and then combine the two branches with concatenation before passing through a dense layer.

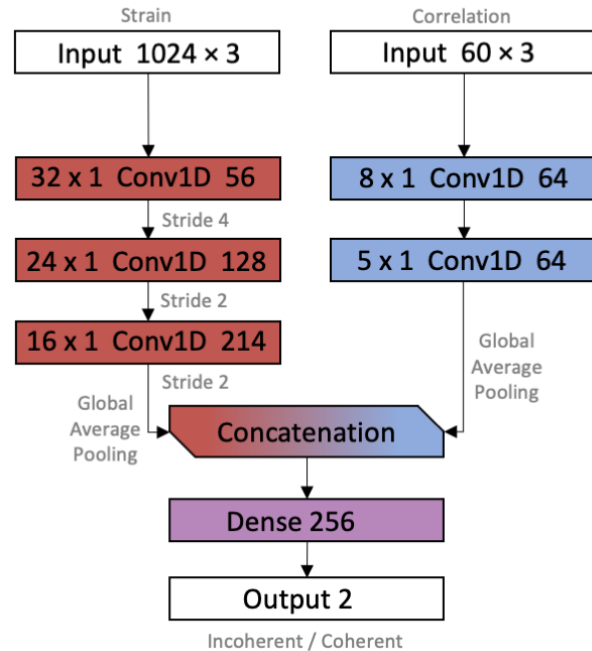


Figure 2. Coherence model architecture (Model 2). The model consist of two branches with separate inputs. The input of the left branch consists of the same three 1024-sample whitened data channels as the input to Model 1 (Figure 1). This is passed through three consecutive convolutional layers and flattened. The input of the right branch consists of the 60-sample Pearson correlation values [equation (1)] between each pair of whitened data streams. This is passed through two convolutional layers, flattened, then concatenated with the output of the strain branch. The combined data is then passed through a dense layer before the output layer. $M \times N$ indicates the size of the filters in each convolutional layer and 64/128/256 is the number of filters (convolutional layers) or nodes (dense layer).

IV. OPTIMIZING THE TRAINING PROCEDURE

A. Measuring performance

A standard means to assess the performance of a GWB detection algorithm (see *e.g.* [77]) is to measure the detection efficiency for various signal morphologies as a function of the signal amplitude (*e.g.*, signal-to-noise ratio) at a fixed false alarm rate (FAR).

We calculate the FAR as a function of score by analysing background data samples that are independent from those used in the training. For the final models we measure the FAR using real LIGO-Virgo data from the second observing run, O2 [78], during times when all three detectors were operating (1 - 25 Aug 2017), with time shifts applied as discussed in Section III A.

We calculate the detection efficiency for a selection of different possible GWB signals, injected in real LIGO-Virgo data from the second observing run. We generate sets of signal injections (different from the ones used for training data) and add them to noise randomly sampled from the O2 observing run during times when all three detectors were operating. We measure our sensitivity to five distinct waveform morphologies:

WNB: These are the same type of signal as the Type I used for training.

CSG: A circularly polarised sinusoidal signal with Gaussian envelope. These *ad hoc* waveforms are standard for testing GWB analyses [77, 78].

CCSN: The N20-2 waveform of [99], from a 3D simulation of a neutrino-driven core-collapse supernova (CCSN) explosion.

Cusp: The GW emission expected from cosmic string cusps [100].

BBH: The GW signal from a black-hole binary merger. We use the IMRPhenomD waveform model [101, 102]. The black-hole masses and spins are selected randomly and uniformly over the intervals $[10, 100] M_{\odot}$ and $[-1, 1]$, with the restriction that the maximum signal frequency does not exceed 512 Hz.

Sample waveforms of each type are shown in Figure 3.

We compute the efficiencies at SNR values $\rho = 0, 1, 2, \dots, 50$, where the $\rho = 0$ case corresponds to pure noise (where we expect approximately zero efficiency). For each SNR value we generate 100 injections. Each injection is rescaled to the desired network SNR, added to the background timeseries, and processed by the models giving a score. The injection is considered to be detected if the combined output score is larger than that of the false alarm rate threshold.

B. Comparing models

It is crucial for a method to have reproducible results if it is going to be compared with another. Machine learning models typically use random initialisation of their trainable parameters, which can lead to different results for repeated runs of the training. As a demonstration of this effect, we train models 1 and 2 ten times each using training data types 0-3, yielding 100 combinations of trained models. Figure 4 shows the distribution of

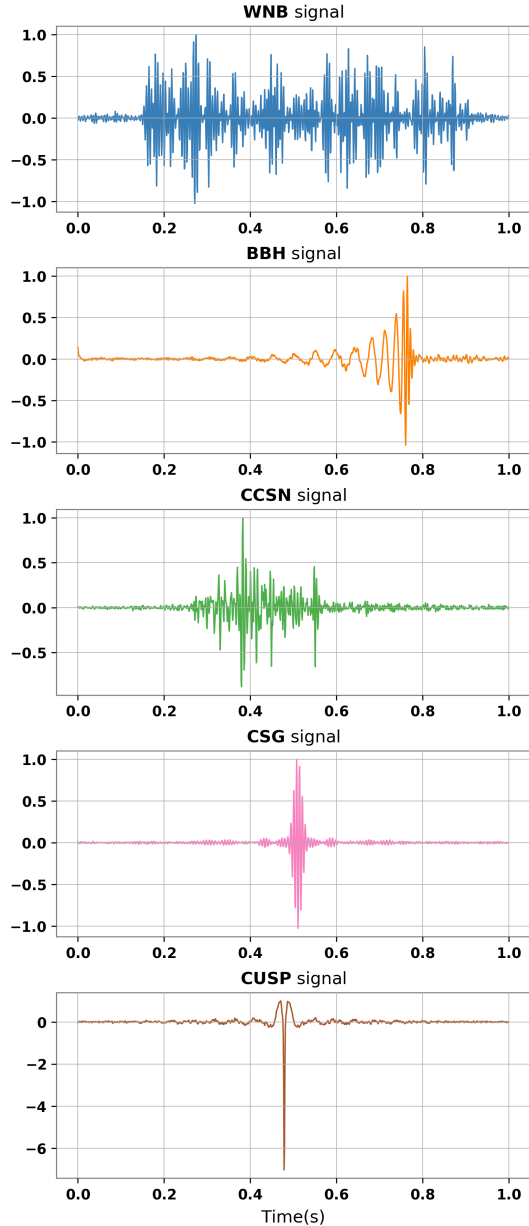


Figure 3. Examples of testing signals: white noise burst (WNB), binary black hole merger (BBH), core-collapse supernova (CCSN), circularly polarised sine-Gaussian (CSG) and cosmic string cusp. All plots show the injection after whitening, but without background noise.

background scores for each trained version of each model

separately and for all 100 model 1 – model 2 pairs. We see that while the performance of each model is variable between trainings, the average performance curves are much smoother and more regular. In the following sections we follow this practice of averaging over repeated trainings of the same model to make more robust comparisons of model performance across different versions of training data.

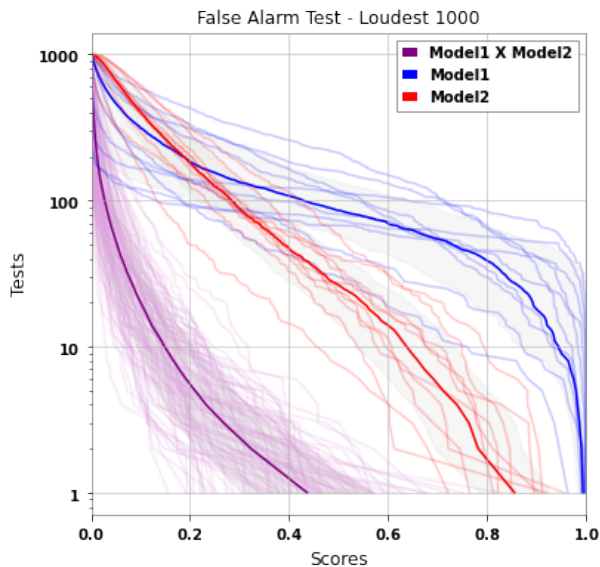


Figure 4. False alarm test results for 10 different trainings of model 1 (blue) and model 2 (red) and their 100 combinations (magenta) of scores. The same 1 month of background noise data was used for testing each trained model. We present here the distribution of the 1000 highest background scores for each model. We highlight with a bold line the mean result for each model. The gray area represent one standard deviation for each case.

C. Training sample ratios

In classification problems the training data often contain the same number of examples in each class, giving all classes equal importance. In our problem we prioritise the reduction of false alarms (false positives) and the ability of the network to recognise noise features. This is motivated by the low false alarm rate threshold adopted by LIGO and Virgo for issuing GWB detection alerts for unmodelled bursts during online running, currently 7.9×10^{-9} Hz (one per 4 years) [103]. Even with our best model, using equal numbers of signal and noise samples we do not get an acceptable performance. Furthermore we have more than just stationary noise and GWB injections in our data types: we also try to simulate single- and multi-detector glitches, so we need to investigate the optimal proportions of each type. We therefore trained both models with different numbers and ratios of the data types to determine which provide the best performance.

For the Coincidence Model (model 1) we tested all combinations of $3N$ and $6N$ instances each of Type 0 and Type 1, and $3N$, $6N$, and $12N$ instances of Type 3, where $N = 10^4$, for a total of 12 different combinations. For the Coherence Model (model 2) we tested all combinations of $5N$ and $10N$ instances each of Type 0, Type 1, Type 2a & b together, and Type 2c, for a total of 16 different combinations. This gives a total of $12 \times 16 = 192$ different combinations of training ratios.

For each combination we repeat the training of each model 7 times and use the mean detection efficiency for comparisons. We evaluate the detection efficiency at fixed false alarm rate of 1/day, using a common set of real background noise data from the second LIGO-Virgo observing run [78] and a common set of GWB injections. The resulting averaged efficiency curve for a given training ratio combination i and waveform w is characterised by the SNRs at which the efficiency reaches 50% and 90%, denoted $\rho_{50\%}^{i,w}$ and $\rho_{90\%}^{i,w}$. We choose as the best training ratio combination that which minimises the following quantity:

$$\chi_i^2 = \sum_w \sum_{E=50\%,90\%} \left(\frac{\rho_E^{i,w} - \min_j[\rho_E^{j,w}]}{\min_j[\rho_E^{j,w}]} \right)^2 \quad (4)$$

Minimising this quantity corresponds to achieving a best average performance (lowest $\rho_{50\%}^{i,w}$ and $\rho_{90\%}^{i,w}$) across all waveforms. We find the best combination to be training model 1 with Types 0, 3, 1 in the amounts $6N$, $6N$, $3N$ and training model 2 with Types 0, 1, 2a&b, 2c in the amounts $10N$, $10N$, $5N$, $5N$. This combination gives amplitude sensitivities ($\rho_{50\%}^{i,w}$ and $\rho_{90\%}^{i,w}$ values) 10%–20% better than training with equal numbers of each type of data. We use these ratios for all training in the rest of this paper.

D. Rescaled Virgo noise level

Up to this point we have trained with artificial Gaussian noise that follows the design noise power spectral density of the advanced LIGO and Virgo detectors [84].

As shown in Fig. 5, when considering the O2 data, the ratio of the true noise level in Virgo to that in the LIGO detectors is higher than the ratio for the design noise spectra used in the training. This means that in the real data we expect the SNR of a signal in Virgo relative to LIGO to be systematically lower than the SNR used in training, potentially resulting in sub-optimal model performance on real data. To investigate this we retrain the best model from the previous investigation (Section IV C) after rescaling the Virgo PSD by the factors 0.5, 1, 2, 4, 8, 16, or 32. Rescaling by 1 is the same training as before, while the factor of 0.5 is for a sanity check on the method. All other factors (> 1) lower the SNR of the training signals in Virgo relative to LIGO. For each case we repeat the training 7 times and repeat the false alarm rate and efficiency measurements of the previous section

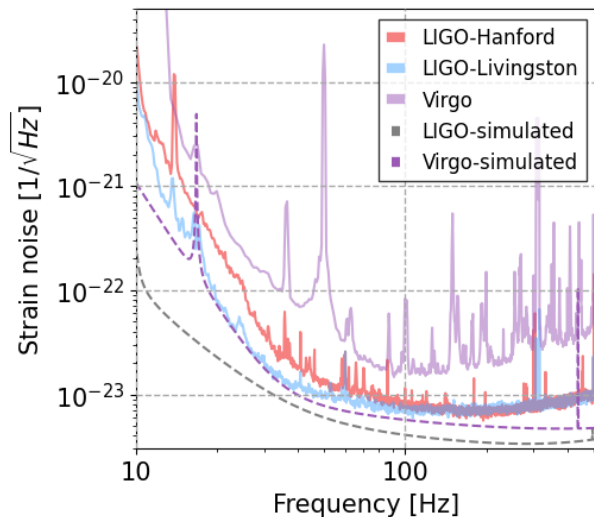


Figure 5. Noise amplitude spectral densities for the LIGO and Virgo detectors during the second observing run, O2 [78]. The dashed lines show the design target sensitivities.

(using new time lags). We then use equation (4) to compare the performance with the different rescalings.

We find a clear performance improvement when training model 1 with rescaling factors of 2 or 4 and model 2 with rescaling factors of 2 – 32. The best performance is found using the rescalings (4,32) for models 1 and 2 respectively, giving typical $\rho_{50\%}^{i,w}$ and $\rho_{90\%}^{i,w}$ values almost a factor of 2 lower than training without rescaling. We use these rescalings for all training in the rest of this paper.

We note with interest that the optimal scaling factor for training model 2 is larger than that for model 1;. We attribute this to the fact that the orientation of the Virgo detector is very different from the near co-alignment of the two LIGO detectors [104]; as a consequence, when training with random timeseries, HV and LV correlations tend to be smaller than HL correlations even for equal PSDs.

Rescaling the Virgo noise level upwards lowers the SNR of training signals in Virgo, causing the models to learn to put less emphasis on the Virgo data. One might question whether this rescaling means that the Virgo data is not useful. As a test, we performed a series of injections where each injection was done twice: once normally, with the signal added to all three detector data streams, and again where the same injections were made into Hanford and Livingston only. We found that while the scores for most injections are largely unchanged, zeroing out the Virgo injection lowers the scores significantly in some cases, particularly for high-SNR signals, while the background distribution is not significantly affected. This demonstrates that even with the rescaling of the Virgo noise for training, the models can still extract useful information from the Virgo data stream.

V. PERFORMANCE OF THE OPTIMALLY TRAINED MODEL

Following our investigations into optimising the training procedure, we now evaluate the performance of MLY’s optimally trained models on real LIGO-Virgo data. For these tests we use models trained with the optimal training sample ratios (Section IV C) and with the optimal Virgo noise rescalings (Section IV D). We use new time lags and new injection sets that haven’t been used previously. Of the 49 combinations of model 1 and model 2 that are trained in this way, we choose the one that has the best performance as measured by equation (4). This best combination is then applied to new time lags and injections for the assessments of performance presented in this section.

Figure 6 shows the detection efficiency of the optimally trained model for our test waveforms (Section IV A) at a FAR of 1/year using real LIGO-Virgo O2 data. We see that MLY is able to detect $>50\%$ ($>90\%$) of all signals that have amplitudes $\rho \geq 19$ ($\rho \geq 27$), with the exception of cusps for which the sensitivity is lower. The performance for CCSN and BBH signals is very similar to that for WNBs, even though the signal morphologies are quite different (see Figure 3). The performance for CSGs is even better, which we attribute to the very small time-frequency volume that it occupies making the signal louder. The performance for cusps is poorer; tests indicate that this occurs because cusps are linearly polarised ($h_{\times} = 0$) while the WNB training injections are unpolarised ($|h_{\times}| = |h_{+}|$). In the next subsection we will see a similar effect for linearly polarised sine-Gaussian signals. This is an interesting demonstration of where our training could potentially be improved to handle a yet broader range of signals.

A. Comparison with the LIGO-Virgo all-sky search in O2

The LIGO-Virgo collaborations have already searched the O2 data looking for generic short-duration GWB signals [78]. We compare the performance of MLY with that of the LIGO-Virgo analysis by repeating the signal injections performed in the LIGO-Virgo search for all signal types reported in [78] with maximum frequencies below 500 Hz; these signal types are listed in Table I and consist of Gaussian pulses, sine-Gaussians, and WNBs (see [78] for details).

The procedure to assess efficiencies is the same as that used in previous sections except that, following LIGO-Virgo convention, instead of using SNR, the injected signal strength is characterised by the root-sum-square value h_{rss} :

$$h_{\text{rss}} = \sqrt{\int_{-\infty}^{\infty} dt (|h_{+}(t)|^2 + |h_{\times}(t)|^2)} \quad (5)$$

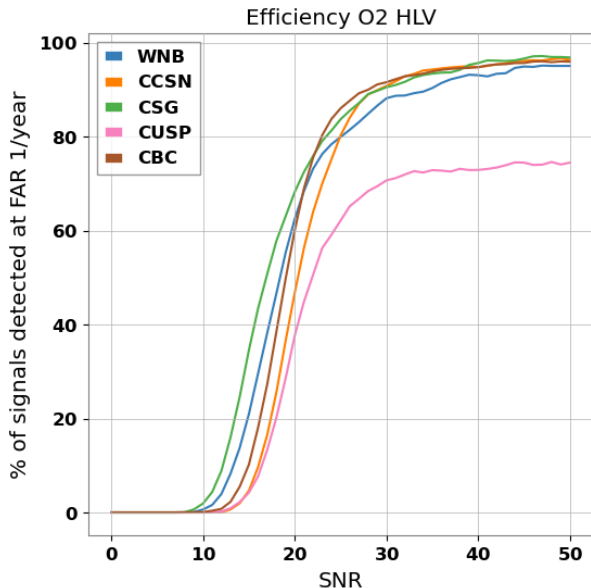


Figure 6. Detection efficiency: the fraction of simulated signals that are detected at a false alarm rate of 1/year versus the network SNR defined by equation (3). The waveform morphologies are white noise burst (WNB), core-collapse supernova (CCSN), circularly polarised sine-Gaussian (CSG), cosmic string cusp (CUSP), and binary black hole merger (BBH). The background noise is real LIGO-Virgo data from O2.

Note that h_{rss} is independent of the detectors or their noise spectra. Hence for a given h_{rss} , signals at frequencies of higher detector noise will have a lower SNR. There is therefore no one-to-one match between SNR and h_{rss} .

Figure 7 shows the detection efficiency of MLY for the tested waveforms, at a FAR threshold of 1/year. We see that most waveforms have similar efficiencies, with only G2D5 and to a lesser extent the SGL153 waveforms having poorer performance. Like the cusp signals in the previous test (Figure 6), the G2D5 and SGL153 waveforms are the only linearly polarised waveforms in the set. Again we conclude that the lower performance is due to training with exclusively unpolarized waveforms.

Table I reports the h_{rss} values at which MLY achieves a detection efficiency of 50% for each waveform. It also shows the h_{rss} values at which the coherent WaveBurst (cWB) [105] pipeline used in the LIGO-Virgo O2 search [78] achieves a detection efficiency of 50% [106]. We see that MLY’s h_{rss} limits are approximately 10% to 50% higher than for cWB, corresponding to sensitivity to distances that are 65% to 90% as far as those of cWB. We consider this to be a very promising first demonstration of the power of machine learning for GWB detection, particularly when one considers the very low computational cost and high speed of the MLY analysis (discussed next).

Name	+/ \times	Parameters	cWB	MLy
($10^{-22} \text{ Hz}^{-1/2}$)				
<i>Gaussian pulses</i>				
G2D5	L	$t=2.5 \text{ ms}$	2.8	3.2
<i>Sine-Gaussian pulses</i>				
SGE70Q3	E	$f_0=70 \text{ Hz}, Q=3$	1.5	1.9
SGE153Q8D9	E	$f_0=153 \text{ Hz}, Q=8.9$	1.3	1.4
SGL153Q8D9	L	$f_0=153 \text{ Hz}, Q=8.9$	–	1.7
SGE235Q100	E	$f_0=235 \text{ Hz}, Q=100$	0.9	1.4
<i>White-Noise Bursts</i>				
WNB100	U	$f_{\text{low}}=100 \text{ Hz},$ $\Delta f=100 \text{ Hz}, t=0.1 \text{ s}$	1.2	1.7
WNB250	U	$f_{\text{low}}=250 \text{ Hz},$ $\Delta f=100 \text{ Hz}, t=0.1 \text{ s}$	1.4	1.7

Table I. Comparison of the detection efficiencies of MLY and cWB at a FAR of 1/year. The first column is the label used for each waveform in Figure 7. The second column indicates the signal polarisation: L (linear: $h_{\times} = 0$), E (elliptical: $|h_{\times}| \leq |h_{+}|$), or U (unpolarised: $|h_{\times}| = |h_{+}|$). The third column lists the waveform parameters; see [78] for definitions. The two rightmost columns are the h_{rss} values in units of $10^{-22} \text{ Hz}^{-1/2}$ at which the MLY and cWB pipelines achieve 50% detection efficiency for each waveform type. A ‘–’ indicates no data available for cWB for that waveform.

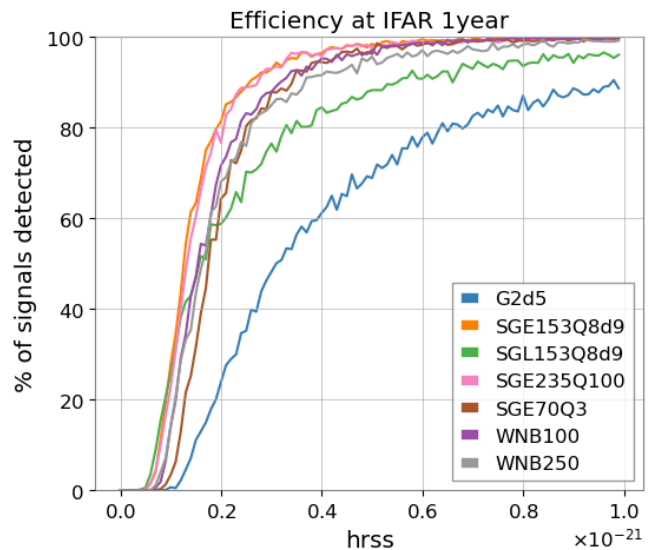


Figure 7. The detection efficiency of the MLY pipeline as a function of the h_{rss} amplitude for the burst signal types listed in Table I. The efficiency is computed at a false alarm rate threshold of 1/year.

B. Inference and Training Times

We note that the CNN analysis of data is very fast: we find that the average time required to process 1 second of whitened data is 51 ms on a 3.5 GHz Xeon E3-1240v5 quad-core CPU with 32 GB of RAM, or approximately 3.3 ms on a A100-SXM4-80GB GPU. This makes second-scale-latency searches feasible, much faster than the minute-scale latency typical of current LIGO-Virgo low-latency unmodelled burst searches [18].

The overall computational cost is also very low compared to traditional search algorithms. The dominant computational cost of our analysis is in estimating the FAR. Approximately 10^3 time shifts is enough to estimate accurately the threshold for FAR values of $O(1/\text{year})$ with days to weeks of data; three or four A100-SXM4 GPUs will therefore be sufficient to perform the full analysis while keeping up with the data in real time. This is in contrast to the several hundred dedicated CPUs typically required by standard algorithms.

The models presented here are the basis of a new pipeline called MLY-Pipeline, that is reviewed and approved to join the last half of the LIGO-Virgo-KAGRA fourth observing run.

Finally, we note that the computational cost of training is modest. A single training of one model on a Tesla V100-SXM2-16GB GPU takes ~ 1 hour, and training both models 7 times each takes ~ 12 hours.

VI. CONCLUSIONS AND FUTURE WORK

We have presented a novel CNN-based analysis pipeline, MLY, for the detection of transient gravitational-wave signals. Unlike previous CNN-based analyses, MLY is capable of detecting waveforms with morphologies that are not included in the training set while rejecting real detector noise glitches. The analysis is shown to be sensitive to a variety of waveform morphologies at signal-to-noise ratios and false alarm rates relevant for issuing rapid alerts to the astronomical community, with very low computing requirements and second-scale latencies possible.

The MLY pipeline uses a multi-component architecture in which one CNN detects transients that are simultaneous in multiple detectors while a second detects correlation between the detectors to eliminate coincident background glitches. The second CNN takes as input both the whitened detector timeseries data and the Pearson correlation between detectors computed for all physically allowed light travel time delays between detectors, allowing the CNN to detect signal correlation rather than

signal shape. We suggest that using separate models to identify different aspects or properties of the desired signal may be a useful approach generally for GW analysis with machine learning methods.

While our model already has sensitivity approaching that of standard low-latency analyses, we consider this investigation to be a promising first attempt with potential for improvement. For example, we could use knowledge of the morphology of common glitch types to reduce the background. Our training used simulated glitches with the same morphology as the simulated GWBs to force our models to recognise GWBs by coincidence and correlation between detectors rather than by signal morphology, since the morphology of real GWBs is not known. However, the morphology of real glitches *is* known. Examination of the noise events in the background distribution shows that many of them are due to known glitch types recognised by Gravity Spy [73, 74, 107, 108]. Applying a glitch classifier or an auto-encoder to candidate events detected by MLY may allow us to identify false alarms as glitches and thereby veto them in real time.

While we have demonstrated MLY’s ability to detect GWBs, the characterisation of any detected signals (commonly referred to “parameter estimation”) is an open problem for unmodelled GWBs, and is necessary for the full exploitation of any detections. Existing machine-learning based methods [23, 27, 28, 56–58] (see also [109]) are fast but rely on precise signal models for training. By contrast the BAYESWAVE algorithm [83] is applicable to generic GWBs, estimating the waveform and providing a map of the probability distribution of the source over the sky, but typically requires hours to run for a single event. Given our goal of using MLY for the low-latency detection of GWBs to allow electromagnetic follow-up observations, a natural next step is to explore how sky-localisation pipelines such as [56] can be generalised to the case of unmodelled GWBs. We leave examination of this topic to a future work.

ACKNOWLEDGMENTS

We thank Erik Katsavounidis for helpful comments on an earlier draft. We thank Shubhanshu Tiwari for providing the sensitivity estimates for cWB listed in Table I. This material is based upon work supported by NSF’s LIGO Laboratory which is a major facility fully funded by the National Science Foundation. The authors are grateful for computational resources provided by the LIGO Laboratory and supported by National Science Foundation Grants PHY-0757058 and PHY-0823459. This work was supported by STFC grants ST/N005430/1 and ST/V005618/1.

[1] J. Aasi, B. Abbott, *et al.* (LIGO Scientific Collaboration), *Classical and Quantum Gravity* **32**, 074001

(2015).

- [2] F. Acernese *et al.* (Virgo Collaboration), *Classical and Quantum Gravity* **32**, 024001 (2014).
- [3] B. Abbott, R. Abbott, T. Abbott, *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), *Physical Review X* **9** (2019), 10.1103/physrevx.9.031040.
- [4] B. P. Abbott, R. Abbott, T. D. Abbott, *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), *The Astrophysical Journal* **892**, L3 (2020).
- [5] R. Abbott, T. Abbott, S. Abraham, *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), *Physical Review D* **102** (2020), 10.1103/physrevd.102.043015.
- [6] R. Abbott, T. D. Abbott, S. Abraham, *et al.*, *The Astrophysical Journal* **896**, L44 (2020).
- [7] R. Abbott, T. D. Abbott, S. Abraham, *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), *Phys. Rev. Lett.* **125**, 101102 (2020).
- [8] <https://gracedb.ligo.org/superevents/public/O3/>.
- [9] T. L. S. Collaboration, the Virgo Collaboration, and the KAGRA Collaboration, “Gwtc-3: Compact binary coalescences observed by ligo and virgo during the second part of the third observing run,” (2021), [arXiv:2111.03606 \[gr-qc\]](https://arxiv.org/abs/2111.03606).
- [10] B. P. Abbott, R. Abbott, T. D. Abbott, *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), *Phys. Rev. Lett.* **116**, 061102 (2016).
- [11] B. P. Abbott, R. Abbott, T. D. Abbott, *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), *The Astrophysical Journal Letters* **848:L12**, 59 (2017), [arXiv:1710.05833 \[astro-ph.HE\]](https://arxiv.org/abs/1710.05833).
- [12] K. Kyutoku, S. Fujibayashi, K. Hayashi, K. Kawaguchi, K. Kiuchi, M. Shibata, and M. Tanaka, *Astrophys. J.* **890**, L4 (2020), [arXiv:2001.04474 \[astro-ph.HE\]](https://arxiv.org/abs/2001.04474).
- [13] B. P. Abbott, R. Abbott, T. D. Abbott, *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), *Phys. Rev. Lett.* **119**, 161101 (2017).
- [14] D. Kasen, B. Metzger, J. Barnes, E. Quataert, and E. Ramirez-Ruiz, *Nature* **551**, 80 (2017).
- [15] B. P. Abbott, R. Abbott, T. D. Abbott, *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), *Phys. Rev. Lett.* **121**, 161101 (2018).
- [16] B. P. Abbott, R. Abbott, T. D. Abbott, *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), *The Astrophysical Journal* **850**, L39 (2017).
- [17] B. P. Abbott, R. Abbott, T. D. Abbott, *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), *Nature* **551**, 85 (2017).
- [18] B. P. Abbott, R. Abbott, T. D. Abbott, *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), *The Astrophysical Journal* **875**, 161 (2019).
- [19] D. George and E. A. Huerta, *Physical Review D* **97**, 044039 (2017), [arXiv: 1701.00008](https://arxiv.org/abs/1701.00008).
- [20] H. Gabbard, M. Williams, F. Hayes, and C. Messenger, *Physical Review Letters* **120**, 141103 (2018), [arXiv: 1712.06041](https://arxiv.org/abs/1712.06041).
- [21] C. Bresten and J.-H. Jung, [arXiv:1910.08245 \[astro-ph, physics:gr-qc, physics:physics\]](https://arxiv.org/abs/1910.08245) (2019), [arXiv: 1910.08245](https://arxiv.org/abs/1910.08245).
- [22] H. Wang, Z. Cao, X. Liu, S. Wu, and J.-Y. Zhu, [arXiv:1909.13442 \[astro-ph, physics:gr-qc\]](https://arxiv.org/abs/1909.13442) (2019), [arXiv: 1909.13442](https://arxiv.org/abs/1909.13442).
- [23] D. George and E. A. Huerta, [arXiv:1711.07966 \[astro-ph, physics:gr-qc\]](https://arxiv.org/abs/1711.07966) (2017), [arXiv: 1711.07966](https://arxiv.org/abs/1711.07966).
- [24] A. Schmitt, K. Fu, S. Fan, and Y. Luo, in *Proceedings of the 2nd International Conference on Computer Science and Software Engineering*, CSSE 2019 (Association for Computing Machinery, New York, NY, USA, 2019) p. 73–78.
- [25] T. D. Gebhard, N. Kilbertus, I. Harry, and B. Schölkopf, *Physical Review D* **100**, 063015 (2019), [arXiv: 1904.08693](https://arxiv.org/abs/1904.08693).
- [26] H.-M. Luo, W. Lin, Z.-C. Chen, and Q.-G. Huang, *Frontiers of Physics* **15**, 14601 (2019).
- [27] X. Fan, J. Li, X. Li, Y. Zhong, and J. Cao, *Science China Physics, Mechanics & Astronomy* **62**, 969512 (2019).
- [28] C. Chatterjee, L. Wen, F. Diakogiannis, and K. Vinsin, “Extraction of binary black hole gravitational wave signals from detector data using deep learning,” (2021), [arXiv:2105.03073 \[gr-qc\]](https://arxiv.org/abs/2105.03073).
- [29] D. S. Deighan, S. E. Field, C. D. Capano, and G. Khanna, *Neural Computing and Applications* (2021), 10.1007/s00521-021-06024-4.
- [30] E. A. Huerta and Z. Zhao, *Handbook of Gravitational Wave Astronomy*, 1–27 (2021).
- [31] W. Wei, A. Khan, E. Huerta, X. Huang, and M. Tian, *Physics Letters B* **812**, 136029 (2021).
- [32] W. Wei and E. Huerta, *Physics Letters B* **816**, 136185 (2021).
- [33] M. L. Chan, I. S. Heng, and C. Messenger, *Phys. Rev. D* **102**, 043022 (2020).
- [34] A. Iess, E. Cuoco, F. Morawski, and J. Powell, “Core-collapse supernova gravitational-wave search and deep learning classification,” (2020), [arXiv:2001.00279 \[gr-qc\]](https://arxiv.org/abs/2001.00279).
- [35] P. Astone, P. Cerdá-Durán, I. Di Palma, M. Drago, F. Muciaccia, C. Palomba, and F. Ricci, *Phys. Rev. D* **98**, 122002 (2018).
- [36] M. López, I. Di Palma, M. Drago, P. Cerdá-Durán, and F. Ricci, *Phys. Rev. D* **103**, 063011 (2021).
- [37] A. W. Steiner, S. Gandolfi, F. J. Fattoyev, and W. G. Newton, *Phys. Rev. C* **91**, 015804 (2015), [arXiv:1403.7546 \[nucl-th\]](https://arxiv.org/abs/1403.7546).
- [38] B. P. Abbott, R. Abbott, Abbott, and et al., *Physical Review Letters* **121** (2018), 10.1103/physrevlett.121.161101.
- [39] A. L. Piro and E. Pfahl, *The Astrophysical Journal* **658**, 1173 (2007).
- [40] M. H. P. M. van Putten, *The Astrophysical Journal* **684**, L91 (2008).
- [41] M. H. P. M. van Putten, N. Kanda, H. Tagoshi, D. Tatum, F. Masa-Katsu, and M. Della Valle, *Phys. Rev. D* **83**, 044046 (2011).
- [42] H. Shen, D. George, E. A. Huerta, and Z. Zhao, (2017), [arXiv: 1711.09919](https://arxiv.org/abs/1711.09919).
- [43] J. McGinn, C. Messenger, M. J. Williams, and I. S. Heng, *Classical and Quantum Gravity* **38**, 155005 (2021).
- [44] C. Chatterjee, L. Wen, F. Diakogiannis, and K. Vinsin, *Phys. Rev. D* **104**, 064046 (2021).
- [45] L. Jiang and Y. Luo, in *2022 26th International Conference on Pattern Recognition (ICPR)* (2022) pp. 46–53.
- [46] E. C. et al., *Machine Learning: Science and Technology* **2**, 011002 (2020).
- [47] E. A. Huerta and Z. Zhao, “Advances in machine and deep learning for modeling and real-time detection of multi-messenger sources,” in *Handbook of Gravitational Wave Astronomy*, edited by C. Bambi, S. Katsanevas, and K. D. Kokkotas (Springer Singapore, Singapore,

- 2020) pp. 1–27.
- [48] E. Cuoco *et al.*, *Mach. Learn. Sci. Tech.* **2**, 011002 (2021), [arXiv:2005.03745](https://arxiv.org/abs/2005.03745) [astro-ph.HE].
- [49] B.-J. Lin, X.-R. Li, and W.-L. Yu, *Frontiers of Physics* **15**, 24602 (2019).
- [50] C. Verma, A. Reza, D. Krishnaswamy, S. Caudill, and G. Gaur, “Employing deep learning for detection of gravitational waves from compact binary coalescences,” (2021), [arXiv:2110.01883](https://arxiv.org/abs/2110.01883).
- [51] J. a. Aveiro, F. F. Freitas, M. Ferreira, A. Onofre, C. m. c. Providência, G. m. c. Gonçalves, and J. A. Font, *Phys. Rev. D* **106**, 084059 (2022).
- [52] A. Menéndez-Vázquez, M. Kolstein, M. Martínez, and L. M. Mir, *Phys. Rev. D* **103**, 062004 (2021).
- [53] C. Ma, S. Wang, W. Wang, and Z. Cao, *Phys. Rev. D* **109**, 043009 (2024).
- [54] S. Jadhav, M. Shrivastava, and S. Mitra, *Machine Learning: Science and Technology* **4**, 045028 (2023).
- [55] M. B. Schäfer, O. c. v. Zelenka, A. H. Nitz, H. Wang, S. Wu, Z.-K. Guo, Z. Cao, Z. Ren, P. Nousi, N. Stergioulas, P. Iosif, A. E. Koloniari, A. Tefas, N. Passalis, F. Salemi, G. Vedovato, S. Klimentko, T. Mishra, B. Brüggemann, E. Cuoco, E. A. Huerta, C. Messenger, and F. Ohme, *Phys. Rev. D* **107**, 023021 (2023).
- [56] C. Chatterjee, L. Wen, K. Vinsen, M. Kovalam, and A. Datta, *Physical Review D* **100** (2019), [10.1103/physrevd.100.103025](https://arxiv.org/abs/10.1103/physrevd.100.103025).
- [57] H. Shen, E. A. Huerta, Z. Zhao, E. Jennings, and H. Sharma, [arXiv:1903.01998](https://arxiv.org/abs/1903.01998) [astro-ph, physics:gr-qc, stat] (2019), [arXiv: 1903.01998](https://arxiv.org/abs/1903.01998).
- [58] H. Gabbard, C. Messenger, I. S. Heng, F. Tonolini, and R. Murray-Smith, *Nature Phys.* **18**, 112 (2022), [arXiv:1909.06296](https://arxiv.org/abs/1909.06296) [astro-ph.IM].
- [59] C. Chatterjee and L. Wen, *The Astrophysical Journal* **959**, 76 (2023).
- [60] S. Khan and R. Green, *Physical Review D* **103** (2021), [10.1103/physrevd.103.064015](https://arxiv.org/abs/10.1103/physrevd.103.064015).
- [61] W. Wei and E. Huerta, *Physics Letters B* **800**, 135081 (2020).
- [62] L. Blanchet, *Living Reviews in Relativity* **17** (2014), [10.12942/lrr-2014-2](https://arxiv.org/abs/10.12942/lrr-2014-2).
- [63] M. Cavaglià, S. Gaudio, T. Hansen, K. Staats, M. Szczepańczyk, and M. Zanolin, *Machine Learning: Science and Technology* **1**, 015005 (2020).
- [64] K. He, X. Zhang, S. Ren, and J. Sun, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016) pp. 770–778.
- [65] S. Sasaoka, Y. Hou, D. S. Dominguez, S. Garg, N. Koyama, Y. Omae, K. Somiya, and H. Takahashi, in *Proceedings of Science* (2023).
- [66] S. Sasaoka, Y. Hou, D. S. Dominguez, S. Garg, N. Koyama, Y. Omae, K. Somiya, and H. Takahashi, in *Proceedings of 38th International Cosmic Ray Conference — PoS(ICRC2023)* (2023).
- [67] E. A. Moreno, B. Borzyszkowski, M. Pierini, J.-R. Vliant, and M. Spiropulu, *Machine Learning: Science and Technology* **3**, 025001 (2022).
- [68] R. Raikman, E. A. Moreno, E. Govorkova, E. J. Marx, A. Gunny, W. Benoit, D. Chatterjee, R. Omer, M. Saleem, D. S. Rankin, M. W. Coughlin, P. C. Harris, and E. Katsavounidis, “Gwak: Gravitational-wave anomalous knowledge with recurrent autoencoders,” (2023), [arXiv:2309.11537](https://arxiv.org/abs/2309.11537) [astro-ph.IM].
- [69] C. D. Ott, *Classical and Quantum Gravity* **26**, 063001 (2009).
- [70] E. Abdikamalov, G. Pagliaroli, and D. Radice, “Gravitational waves from core-collapse supernovae,” in *Handbook of Gravitational Wave Astronomy*, edited by C. Bambi, S. Katsanevas, and K. D. Kokkotas (Springer Singapore, Singapore, 2020) pp. 1–37.
- [71] V. Skliris, M. R. K. Norman, and P. J. Sutton, (2020), [arXiv:2009.14611v1](https://arxiv.org/abs/2009.14611v1) [astro-ph.IM].
- [72] T. Marianer, D. Poznanski, and J. X. Prochaska, *Monthly Notices of the Royal Astronomical Society* **500**, 5408 (2020), <https://academic.oup.com/mnras/article-pdf/500/4/5408/34925881/staa3550.pdf>.
- [73] M. Zevin, S. Coughlin, S. Bahaadini, E. Besler, N. Rohani, S. Allen, M. Cabero, K. Crowston, A. K. Katsaggelos, S. L. Larson, T. K. Lee, C. Lintott, T. B. Littenberg, A. Lundgren, C. Østerlund, J. R. Smith, L. Trouille, and V. Kalogera, *Classical and Quantum Gravity* **34**, 064003 (2017).
- [74] S. Bahaadini, V. Noroozi, N. Rohani, S. Coughlin, M. Zevin, J. Smith, V. Kalogera, and A. Katsaggelos, *Information Sciences* **444**, 172 (2018).
- [75] M. J. Szczepańczyk, F. Salemi, S. Bini, T. Mishra, G. Vedovato, V. Gayathri, I. Bartos, S. Bhaumik, M. Drago, O. Halim, C. Lazzaro, A. Miani, E. Milotti, G. A. Prodi, S. Tiwari, and S. Klimentko, *Phys. Rev. D* **107**, 062002 (2023).
- [76] S. Klimentko *et al.*, *Phys. Rev. D* **93**, 042004 (2016), [arXiv:1511.05999](https://arxiv.org/abs/1511.05999) [gr-qc].
- [77] B. P. Abbott, R. Abbott, T. D. Abbott, *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), *Phys. Rev. D* **95**, 042003 (2017).
- [78] B. Abbott, R. Abbott, T. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, R. Adhikari, V. Adya, C. Affeldt, and *et al.*, *Physical Review D* **100** (2019), [10.1103/physrevd.100.024017](https://arxiv.org/abs/10.1103/physrevd.100.024017).
- [79] B. P. Abbott, R. Abbott, T. D. Abbott, *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), *Classical and Quantum Gravity* **33**, 134001 (2016).
- [80] S. Klimentko, G. Vedovato, V. Necula, F. Salemi, M. Drago, E. Chassande-Mottin, V. Tiwari, C. Lazzaro, B. O’Brian, M. Szczepańczyk, S. Tiwari, and V. Gayathri, (2020).
- [81] R. Lynch, S. Vitale, R. Essick, E. Katsavounidis, and F. Robinet, *Phys. Rev. D* **95**, 104046 (2017).
- [82] P. J. Sutton, G. Jones, S. Chatterji, P. Kalmus, I. Leonor, S. Poprocki, J. Rollins, A. Searle, L. Stein, M. Tinto, and M. Was, *New Journal of Physics* **12**, 053034 (2010).
- [83] N. J. Cornish and T. B. Littenberg, *Classical and Quantum Gravity* **32**, 135012 (2015).
- [84] B. P. Abbott, R. Abbott, T. D. Abbott, *et al.* (LIGO Scientific Collaboration and Virgo Collaboration and KAGRA Collaboration), *Living Rev Relativ* **21** (2013), [arXiv:1304.0670](https://arxiv.org/abs/1304.0670) [gr-qc].
- [85] <https://www.gw-openscience.org/>.
- [86] J. W. Murphy, C. D. Ott, and A. Burrows, *The Astrophysical Journal* **707**, 1173 (2009).
- [87] T. Damour and A. Vilenkin, *Phys. Rev. Lett.* **85**, 3761 (2000).
- [88] V. Skliris, “MLy, gravitational waves and machine learning library,” <https://git.ligo.org/vasileios.skliris/mly>.

- [89] A. Nitz, I. Harry, D. Brown, C. M. Biwer, J. Willis, T. D. Canton, *et al.*, “[gwastro/pycbc: Pycbc release v1.16.9](#),” (2020).
- [90] D. Macleod, A. L. Urban, S. Coughlin, T. Massinger, M. Pitkin, R. George, Paulaltin, J. Areeda, L. Singer, E. Quintero, K. Leinweber, and T. G. Badger, “[gwpwpy/gwpwpy: 2.0.1](#),” (2020).
- [91] F. Chollet *et al.*, “[Keras](#),” (2015).
- [92] <https://github.com/VasSkiris/mly-methods-paper>.
- [93] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, [Data Mining and Knowledge Discovery](#) **33**, 917–963 (2019).
- [94] M. R. K. Norman, *Evolving Attention*, Ph.D. thesis, Cardiff University (2024).
- [95] L. N. Smith, “Cyclical learning rates for training neural networks,” (2017), [arXiv:1506.01186 \[cs.CV\]](#).
- [96] L. Cadonati, [Classical and Quantum Gravity](#) **21**, S1695–S1703 (2004).
- [97] L. Cadonati and S. Márka, [Classical and Quantum Gravity](#) **22**, S1159 (2005).
- [98] S. Chatterji, A. Lazzarini, L. Stein, P. J. Sutton, A. Searle, and M. Tinto, [Phys. Rev. D](#) **74**, 082005 (2006).
- [99] Müller, E., Janka, H.-Th., and Wongwathanarat, A., [A&A](#) **537**, A63 (2012).
- [100] T. Damour and A. Vilenkin, [Phys. Rev. D](#) **64**, 064008 (2001).
- [101] S. Husa, S. Khan, M. Hannam, M. Pürrer, F. Ohme, X. J. Forteza, and A. Bohé, [Phys. Rev. D](#) **93**, 044006 (2016).
- [102] S. Khan, S. Husa, M. Hannam, F. Ohme, M. Pürrer, X. J. Forteza, and A. Bohé, [Phys. Rev. D](#) **93**, 044007 (2016).
- [103] <https://emfollow.docs.ligo.org/userguide/>.
- [104] A simple measure of the overlap of detectors α and β is $O_{\alpha\beta} \equiv 2\text{Tr}(\mathbf{D}_\alpha\mathbf{D}_\beta) \in [-1, 1]$, where \mathbf{D} is the detector response tensor [110]. One finds $O^{HL} = -0.89$, $O^{LV} = -0.25$, and $O^{VH} = -0.016$.
- [105] S. Klimenko, G. Vedovato, M. Drago, F. Salemi, V. Tiwari, G. A. Prodi, C. Lazzaro, K. Ackley, S. Tiwari, C. F. Da Silva, and G. Mitselmakher, [Phys. Rev. D](#) **93**, 042004 (2016), [arXiv:1511.05999 \[gr-qc\]](#).
- [106] S. Tiwari, private communication.
- [107] S. B. Coughlin, S. Bahaadini, N. Rohani, M. Zevin, O. Patane, M. Harandi, C. Jackson, V. Noroozi, S. Allen, J. Areeda, M. W. Coughlin, P. Ruiz, C. P. L. Berry, K. Crowston, A. K. Katsaggelos, A. Lundgren, C. Osterlund, J. R. Smith, L. Trouille, and V. Kalogera, [Physical Review D](#) **99**, 082002 (2019), [arXiv: 1903.04058](#).
- [108] S. Soni *et al.*, [Class. Quant. Grav.](#) **38**, 195016 (2021), [arXiv:2103.12104 \[gr-qc\]](#).
- [109] L. P. Singer and L. R. Price, [Physical Review D](#) **93** (2016), [10.1103/physrevd.93.024013](#).
- [110] W. G. Anderson, P. R. Brady, J. D. E. Creighton, and E. E. Flanagan, [Phys. Rev. D](#) **63**, 042003 (2001).