From One to Many: A Deep Learning Coincident Gravitational-Wave Search

Marlin B. Schäfer[1,2], Alexander H. Nitz[1,2]
[1] *Max-Planck-Institut für Gravitationsphysik, Albert-Einstein-Institut, D-30167 Hannover, Germany and*
[2] *Leibniz Universität Hannover, D-30167 Hannover, Germany*

Gravitational waves from the coalescence of compact-binary sources are now routinely observed by Earth bound detectors. The most sensitive search algorithms convolve many different pre-calculated gravitational waveforms with the detector data and look for coincident matches between different detectors. Machine learning is being explored as an alternative approach to building a search algorithm that has the prospect to reduce computational costs and target more complex signals. In this work we construct a two-detector search for gravitational waves from binary black hole mergers using neural networks trained on non-spinning binary black hole data from a single detector. The network is applied to the data from both observatories independently and we check for events coincident in time between the two. This enables the efficient analysis of large quantities of background data by time-shifting the independent detector data. We find that while for a single detector the network retains 91.5% of the sensitivity matched filtering can achieve, this number drops to 83.9% for two observatories. To enable the network to check for signal consistency in the detectors, we then construct a set of simple networks that operate directly on data from both detectors. We find that none of these simple two-detector networks are capable of improving the sensitivity over applying networks individually to the data from the detectors and searching for time coincidences.

I. INTRODUCTION

Gravitational waves (GWs) are now routinely observed by the two Advanced LIGO detectors [1] and the Advanced Virgo detector [2]. At the end of the last observing period, the KAGRA detector [3] joined the network and is expected to aid observations in the future. During three observing runs $\approx$ 60 GWs from compact binary sources have been identified, almost all of which are consistent with the merger of binary black hole (BBH) systems [4–7].

Many searches for GWs from compact-binary coalescence use matched filtering to separate potential signals from the background detector noise [5, 8–10]. Matched filtering is a technique that convolves a set of pre-calculated template waveforms, each representing a possible source with different component masses, spins, etc., with the detector's data and is known to be optimal for Gaussian noise [11]. A signal-to-noise ratio (SNR) time series is calculated for each template waveform; candidates are identified by a peak in the SNR time series that also passes data quality [12–14] checks. In a second step the candidate detections from one detector are cross-validated with the candidate detections from other detectors to further increase the significance of the reported events and rule out false positives [5, 15, 16]. For sources where the gravitational-wave signal is unknown or poorly modelled other search algorithms detect coincident excess power in different detectors and don't require a model [17].

Deep learning has started to be explored as an alternative approach to building an algorithm to detect GWs [18–27]. It may potentially target signals which are currently challenging for matched filter search algorithms due to computational limitations [24, 28]. The computational cost of these modeled searches scales with the number of templates required by the parameter space. Certain effects like higher-order modes [29], precession [30], eccentricity [31, 32], or the inclusion of sub-solar mass systems [33, 34] potentially require millions of templates and are thus computationally prohibitive to analyze. Deep learning may also be more sensitive when the noise is non-Gaussian [25, 35, 36].

In our previous work [37] we explored the sensitivity of a simple neural network to non-spinning BBH sources in Gaussian noise for a single detector. We tested how different training strategies influence the training procedure and the final efficiency of the network. Our results showed that under the given conditions the network can closely reproduce the sensitivity of matched filtering and that most efficient convergence is reached when a range of low SNR signals is provided throughout training.

Here we extend our previous work to two detectors. To do so, we use the same single detector network explored in [37] and apply it individually to the data from both observatories. This procedure produces a list of candidate events for each detector. We then search for coincident events between the two, where two events are assumed to be coincident if they are within the maximum time-of-flight difference between both detectors. We assume this difference to be 0.1 s since the networks are trained to be insensitive to variations on such scale.

The network uses the unbounded Softmax replacement (USR) modification we introduced in [37]. It outputs a single detector ranking statistic. Here we use it to construct a network ranking statistic. This network ranking statistic turns out to be the sum of the individual ranking statistics minus a correction factor.

The main advantage of this approach is the trivial computation of the search background which enables robust detection claims at comparable statistical significance ($<$ 1 per 100 years) to existing production method-

ology. By applying time shifts larger than the time-of-flight difference between the detectors to the data from only one observatory, we can create large amounts of data which by construction cannot contain any astrophysical coincident candidates. By applying the time shifts to the single detector events rather than the input data directly, we can skip re-evaluating the entire test set and efficiently look for coincident events. This is a well established method that has already been successfully applied [5, 15, 16]. By this approach we can probe the search down to a false-alarm rate (FAR) of 1 false-alarm per $\mathcal{O}\left(10^3\right)$ months. The FAR estimates how often a candidate is produced by the search under the null hypothesis of no astrophysical candidates. Our FAR-estimate is limited by the assigned hardware resources rather than the available data.

We compare this search to an equivalent matched filter search [38]. We find that the deep learning search still retains 92.4% of the sensitivity of a two-detector matched filter search when the latter is restricted to using the timing difference between the detectors as the only means for determining coincident events. However, the matched filter search also extracts some information on the parameters of the signal. When we also require matching templates and the phase and amplitude of the triggered templates to be consistent between detectors [39], the machine learning search only retains 83.9% of the sensitivity.

We then construct a single network that operates on the data from both detectors. The idea is that the network may then be able to learn, summarize, and cross-correlate signal characteristics between detectors. To do so, we remove the last layer of the original networks applied to the individual detectors and concatenate their output. Thereby the input data are compressed to a 128 dimensional latent space. Dense layers are used to correlate the concatenated outputs and condense it into a single ranking statistic.

Using a single network complicates the background estimation, as time shifts between the detectors can in principle not be applied after evaluating the individual data streams. However, the two-detector network architecture is constructed such that the data from different detectors is analyzed by individual sub-networks, concatenated and processed by a third sub-network. This enables us to process the bulk of the data only once and apply time shifts to the individual detector sub-network outputs. To obtain the ranking statistic we are then only required to run the time-shifted data through the final, small sub-network.

We find that networks constructed this way are not able to improve the sensitivity over a time coincidence analysis of the single detector machine learning events. We test three different approaches to training these networks but none show any improvement.

TABLE I. A detailed overview of the architecture for the single detector neural network. Rows are grouped by their influence on the shape of the data. The layers are to be read from left to right and top to bottom to construct the network.

layer type	kernel size	output shape
Input + BatchNorm1d		2048×1
Conv1D + ELU	64	1985×8
Conv1D	32	1954×8
MaxPool1D + ELU	4	488×8
Conv1D + ELU	32	457×16
Conv1D	16	442×16
MaxPool1D + ELU	3	147×16
Conv1D + ELU	16	132×32
Conv1D	16	117×32
MaxPool1D + ELU	2	58×32
Flatten		1856
Dense + Dropout + ELU		64
Dense + Dropout + ELU		64
Dense + Softmax		2

II. COINCIDENT SEARCH FROM INDEPENDENT SINGLE-DETECTOR NETWORKS

The algorithm explored in this section uses a network trained on data from a single detector and uses it to find coincidences in multiple detectors. It is one of the most simple extensions and has two advantages. Firstly, networks trained on data from a single detector can be re-used which reduces requirements to computational resources. Secondly, the search background can be estimated using well established and efficient algorithms allowing for much higher confidence in candidate detections.

A. Architecture

We use the same network as in [37], which is an adaptation of the network presented in [20]. It consists of 6 stacked convolutional layers followed by 3 dense layers. An overview of the architecture is given in Table I.

The last layer contains a Softmax activation function, which we remove during testing. In [37] we showed that this modification, which we called unbounded Softmax replacement (USR), allows the network to be tested at lower FARs than otherwise possible.

The Softmax activation for the first output neuron is given by

$$p \coloneqq \text{Softmax}\left(\boldsymbol{x}\right)_0 = \frac{1}{1+\exp\left(-\Delta x\right)}, \tag{1}$$

where $\boldsymbol{x} = (x_0, x_1)$ is the network output before the activation function and $\Delta x = x_0 - x_1$. When Δx is strongly positive, the denominator in (1) and thus the fraction numerically evaluates to 1. This leads to problems when

setting the threshold value to use to determine true positive detections [37].

However, equation (1) is bijective and can be inverted

$$-\Delta x = \log\left[\frac{1}{p} - 1\right]. \tag{2}$$

This quantity is monotonic and we can thus do statistics on Δx directly, avoiding numerical instabilities while still using the Softmax activation during training.

B. Data Sets and Training

The input to the network is a time series of 1 s duration sampled at 2048 Hz. This allows for signals up to a frequency of 1024 Hz to be resolved which is sufficient for the considered parameter space.

The network is trained on signals from non-spinning BBHs with component masses m_1, m_2 uniformly distributed from $10\,\mathrm{M}_\odot$ to $50\,\mathrm{M}_\odot$. We enforce $m_1 \geq m_2$ and for each pair of masses uniformly draw 5 coalescence phases $\phi_0 \in [0, 2\pi]$. The signals are generated with the waveform model `SEOBNRv4_opt` [40] (optimized version of `SEOBNRv4` [41]) and scaled to varying optimal SNRs in the range $[5, 15]$ during training. The time of merger is varied from 0.6 s to 0.8 s from the start of the input window to decrease the dependency of the network on the exact signal position. Each signal is whitened by the analytic model for the detector power spectral density (PSD) `aLIGOZeroDetHighPower` [42]. For further details on the training set please refer to [37].

Notably, we do not vary the sky position, inclination or polarization during training. For a single detector, variations in these parameters can be fully expressed by changes in the distance, which is fixed by choosing a specific SNR, and the phase ϕ_0. For a two detector setup this degeneracy is broken as a time-of-flight difference is introduced and the amplitudes and phases are correlated in the two detectors. However, our search algorithm is largely parameter agnostic. This means that its output does not depend on the amplitude or phase. Thus, we do not have information on whether or not the search responds to consistent signals. Finally, the time-of-flight difference is on the order of the variation of the merger time within the training set and can, therefore, not be resolved. In section III the network has access to data from both observatories and the data is adjusted accordingly.

All noise is Gaussian and simulated from the `aLIGOZeroDetHighPower` PSD [42]. We explicitly generate colored noise and whiten it afterwards. This in principle allows to extend our training to real noise.

The training set contains 200 000 noise samples, 100 000 of which are combined with 100 000 unique signals. The validation set[1] contains 400 000 noise samples and 10 000 unique signal samples, which we subsequently scale to SNRs $3, 6, 9, 12, 15, 18, 21, 24, 27$ and 30. This set is used to calculate the *efficiency* of the network at a fixed false-alarm probability (FAP) of 10^{-4}. The FAP is the fraction of discrete noise samples misclassified as signals. The efficiency is the fraction of discrete signal samples correctly classified as signals at a given FAP.

The test set contains a month of continuous simulated noise for each of the two detectors in Hanford and Livingston. We inject signals with parameters drawn from the distributions shown in Table II into both data streams. Injections are separated by a random time between 16 s to 22 s. To enable the networks to process this data, the continuous stream is sliced into ≈ 26 million overlapping, correlated samples. Each sample is whitened individually by the analytic PSD.

We construct a second test set for background estimation. This set contains the same time domain noise as the first test set but no injections are performed. We pre-process this second data set in the same way we pre-process the first data set for the network to be able to process it.

The network is trained for 200 epochs and we use the network with the highest average efficiency over all SNRs for the analysis carried out here. We use the Adam optimizer with a learning rate of 10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ [43]. We use a variant of the binary cross-entropy which was designed to stay finite as loss function

$$L(\boldsymbol{y}_\mathrm{t}, \boldsymbol{y}_\mathrm{p}) = -\frac{1}{N_\mathrm{b}} \sum_{i=1}^{N_\mathrm{b}} \boldsymbol{y}_{\mathrm{t},i} \cdot \log\left(\epsilon + (1 - 2\epsilon)\boldsymbol{y}_{\mathrm{p},i}\right), \tag{3}$$

where $\boldsymbol{y}_\mathrm{t}$ is $(1, 0)^T$ for a signal-class sample and $(0, 1)^T$ for a noise-class sample, $\boldsymbol{y}_\mathrm{p}$ is the prediction of the network, $N_b = 32$ is the mini-batch size, and $\epsilon = 10^{-6}$.

We implemented the network using the high-level API Keras [44] of TensorFlow version 2.3.0 [45].

C. Single Detector Events

To apply the network to data of duration longer than the 1 s input of the network, we use a sliding window with step size 0.1 s. The contents of each window are whitened individually by the PSD model. At each step the network outputs a set of two numbers, the difference of which we use as our ranking statistic.

We apply the same network to the data from both detectors individually. We, thus, receive two output time series of ranking statistics. To determine notable events in the individual detectors we apply a threshold to both time series and cluster the resulting points above the threshold into events. A point exceeding the threshold is

[1] In out previous work [37] what we call validation set here was named efficiency set.

counted towards a cluster if it is within 0.2 s of the cluster boundaries. We choose a threshold on the USR output of -2.2, which corresponds to a Softmax output of 0.1.

The search algorithm produces a list of events, where an event is a tuple $(t, \Delta x)$. Each event is a time t at which the network predicts a signal to be present with a ranking statistic Δx. The ranking statistic can be used to assign a significance to the event.

D. Coincident Events

A signal will be present in the data of all detectors if it is of astrophyiscal origin. Its SNR in each detector depends on the location and orientation of the source. The number of false alarms can, thus, be reduced by requiring that the event is picked up by multiple detectors at similar times.

To quantify the significance of an event detected by more than one observatory, a combined ranking statistic is required. For simplicity we restrict our current analysis to two detectors. However, this approach is extendable to any number of detectors.

If the network was using the final Softmax activation during evaluation a combined ranking statistic would come straight forwardly from the interpretation of the output as a probability.

$$p_{H+L} = 1 - (1 - p_H)(1 - p_L) \tag{4}$$

The 1-to-1 relation between p and Δx given in equation (1) can be inserted into (4) to get

$$\begin{aligned} -\Delta x_{H+L} = &- \Delta x_H - \Delta x_L \\ &- \log\left[1 + e^{-\Delta x_H} + e^{-\Delta x_L}\right]. \end{aligned} \tag{5}$$

The combined ranking statistic is the sum of the single detector ranking statistics minus a correction term.

We consider an event in one detector to be coincident with another event in the other detector if the event times t_i are within 0.1 s of each other. This time difference is chosen to be the maximum time resolution the networks can achieve due to the time variation in the training set.

We construct a list of coincidence events from the single detector list by the above condition. Each coincident event is assigned the combined ranking statistic (5) and the time in the Hanford detector.

E. Background Estimation

To estimate the FAR at different ranking statistic values we evaluate the same noise used to search for signals but omit injecting the GWs. This ensures that all events found in this data set are noise artifacts and are not influenced by close by injections.

We apply the network to the data and determine events as described in section II C. We obtain two lists of events and search for coincidences as detailed in section II D.

The lowest FAR that can be probed is limited by the duration of the analyzed data. Our test set covers one month. The duration can be increased by shifting the data in one of the detectors by a time larger than the maximum time-of-flight duration between the detectors. Rather than shifting the data itself one may instead alter the event times returned by the search. This allows us to skip reanalyzing the full data for each time step and only requires us to look for coincidences between the events from one detector and the time shifted events from the second detector. Increasing the amount of background by applying time shifts is a well established method that has already been successfully applied in production searches [5, 15, 16].

We choose a time shift of 1024 s and apply any possible integer multiple of this step size. We then search for coincidences in these events as detailed in section II D. This procedure increases our background to ≈ 2400 months $=$ 200 years.

A list of FARs at different network ranking statistics is obtained by counting the number of events in the way described above with a larger ranking statistic.

F. Sensitivity

The sensitive volume of a search can be estimated by

$$V(\mathcal{F}) \approx V(d_{\text{max}}) \frac{N_s(\mathcal{F})}{N_{\text{inj}}}, \tag{6}$$

when it is derived on data containing injections which are distributed uniformly in volume [15]. Here $\mathcal{F}$ is the FAR at which the volume is being calculated, d_{max} is the maximum distance of any injection, $V(d_{\text{max}})$ is the volume of a sphere with radius d_{max}, $N_s(\mathcal{F})$ is the number of signals detected with a FAR $\leq \mathcal{F}$ and N_{inj} is the total number of injected signals. We report the radius of a sphere with volume $V(\mathcal{F})$ instead of the sensitive volume.

We analyze a month of simulated data from the two detectors Hanford and Livingston, assuming the PSD `aLIGOZeroDetHighPower` [42]. The data contains injections drawn from the distribution shown in Table II. We apply the network to the data from both detectors individually as described in section II C. The resulting single detector events are correlated and a list of coincident events is produced as detailed in section II D. We then pick out any events that are within 0.3 s of an injection. These events are called foreground events from here on out.

To determine the search background, we evaluate the same month of noise used to find the foreground events. However, this data does not contain any injections. The networks return a list of single detector events, which are correlated and shifted in time to increase the effective duration of the analyzed data as detailed in section II E. The resulting coincident events are called background events from here on out.

TABLE II. Distributions of the parameters used for the injections in the test set.

Parameter	Uniform distribution
Component masses	$m_1, m_2 \in (10, 50)\,\mathrm{M}_\odot$
Spins	0
Coalescence phase	$\Phi_0 \in (0, 2\pi)$
Polarization	$\Psi \in (0, 2\pi)$
Inclination	$\cos\iota \in (-1, 1)$
Declination	$\sin\theta \in (-1, 1)$
Right ascension	$\varphi \in (-\pi, \pi)$
Distance	$\mathrm{d}^2 \in \left(500^2, 7000^2\right)\,\mathrm{Mpc}^2$

We can then assign a FAR to any foreground event. To do so we count the number of background events with a ranking statistic larger than the ranking statistic of the considered foreground event. This number is divided by the effective duration of the analyzed background to obtain a FAR. The sensitive volume is then obtained from equation (6) and converted to a distance. The sensitive distance as a function of the FAR is obtained by evaluating the sensitive volume at the FARs of all foreground events.

G. Matched Filtering

The template bank contains 598 unique waveforms and is constructed such that no more than 3% of the SNR of any signal is lost due to the discreteness of the bank. It covers the same mass range of $10\,\mathrm{M}_\odot$ to $50\,\mathrm{M}_\odot$ as the training set of the networks and spins are set to 0. The individual templates are generated using the waveform model `IMRPhenomD` [46, 47] and placed stochastically.

To run the matched filter search we use the program `pycbc_inspiral` [38]. It is setup to use a SNR threshold of 5 in both detectors to create two sets of single detector triggers. These two sets are then checked for coincidence by two different approaches.

One approach handles the matched filter triggers analogous to the network single detector triggers, i.e. they are clustered and turned into single detector events as described in section II C. In this case the ranking statistic is the SNR returned by the best matching template. We then look for coincidences as described in section II D by requiring two events in different detectors to be separated by no more than 0.1 s. The combined ranking statistic in this case is given by

$$\rho_{H+L} = \sqrt{\rho_H^2 + \rho_L^2}. \tag{7}$$

This disregards the information about the possible parameters obtained from the best matching template and only looks for time coincidence, i.e. no signal consistency is required.

The other approach leverages the signal information and checks for phase and amplitude correlation as well as requiring that the templates matching the data are consistent between detectors. In particular we utilize the combined ranking statistic given in equation (2) of [39] and find coincidences as described therein.

H. Evaluation and Comparison to Matched Filtering

In Figure 1 we show the injections that were found and missed by the network coincident search at a FAR of 1 false alarm per month. The x-axis shows the optimal SNR of the injections in the Hanford detector and the y-axis shows the optimal SNR in the Livingston detector. The color indicates the network ranking statistic as calculated by equation (5). Missed injections are marked with a red cross. A network SNR of 8 as calculated by equation (7) is highlighted by the black line.

Figure 1 shows that the combined ranking statistic (5) is correlated with the network SNR. As the network SNR increases so does the combined ranking statistic. The loudest missed injection has a network SNR of 22.7. However, the signal is most dominantly seen in the Hanford detector with a single detector SNR of 22.6, whereas Livingston has an optimal SNR < 2 due to the location of the source. Therefore, it is not surprising that the signal does not show up in both detectors and is missed by the coincidence search. When considering only the detector in which the signal is observable with lower SNR, the loudest missed signal has a optimal SNR of 9.2 in that detector.

In Figure 2 we show the sensitive distance of different algorithms as a function of the FAR. The orange lines show the sensitivity curves of the machine learning based algorithms whereas the purple lines show the sensitivities of a comparable matched filter search. The dashed lines show the sensitivity of the searches when only a single detector is considered. We compare those to a two-detector search where we require coincident detections in both detectors. The filled orange line and the dash-dotted purple line show the comparison between the machine learning and matched filter algorithms, respectively, when both impose the same coincidence condition. The filled purple line shows a more realistic application of matched filtering where the consistency of the time of arrival, the phase, the amplitude, as well as the parameters of the best matching template are required.

We find a significant improvement of up to 20% at a given FAR when the machine learning algorithm has access to data from both detectors compared to using only data from a single detector. Furthermore, we can probe FARs down to $\approx 4 \times 10^{-4}$ false alarms per month without needing to increase the amount of evaluated data by applying time shifts between detectors as described in section II E. In principle this limit may be decreased even further and time shifts are only limited by the time-of-flight difference between the detectors. The large increase in the available background potentially greatly increases

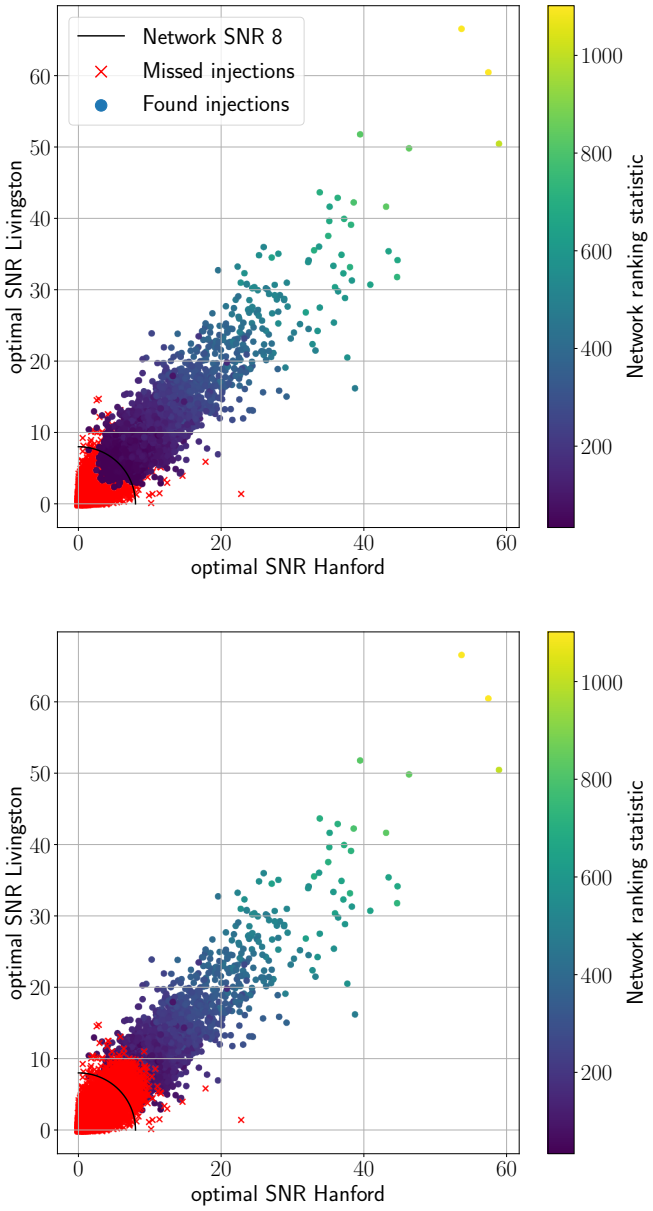

FIG. 1. Found and missed injections from the test set as returned by the procedure discussed in section II. The top panel overlays the missed injections by the found injections and the bottom panel reverses the order. The x- and y-axis show the optimal SNRs of the injections in the Hanford and Livingston detector, respectively. The color of found injections represents the combined ranking statistic as defined by equation (5). Missed injections are marked by a red cross. The black line indicates an optimal network SNR of 8. The plot is generated at a FAR of 1 false alarm per month.

the statistical significance of any event.

The sensitivities of the machine learning search algorithms are compared to an equivalent matched filter search. For the single detector searches given by the dashed lines in Figure 2 we find that the machine learning algorithm retains at least 91.5% of the sensitivity at a fixed FAR of the matched filter analogue. This corresponds to a maximum absolute separation of 200 Mpc. This difference in sensitivity is basically unchanged when data from two detectors is considered and both the machine learning as well as the matched filter search calculate coincidences only based on the timing in the different detectors. The corresponding curves in Figure 2 are the filled orange and the dash-dotted purple line, respectively. In this case, the machine learning algorithm retains at least 92.4% of the sensitivity of the time coincidence matched filter search which corresponds to an absolute separation of 180 Mpc.

However, matched filtering also carries information about the intrinsic parameters of the source, the relative phase, and the relative amplitudes in the two detectors. This information can be used to further constrain coincidences and improve the ranking statistic [39] by testing for signal consistency. We compare the time coincidence machine learning search (filled, orange line in Figure 2) to this matched filter coincidence search utilizing signal consistency checks (filled, purple line in Figure 2). The machine learning search now only retains at least 83.9% of the sensitivity in FAR regions where both are defined. This corresponds to an absolute separation of 430 Mpc.

We truncate the sensitivity curve of any search that has access to data from both detectors in Figure 2 at a FAR of 10^3 false alarms per month. This is done due to a large number of true positives at high FARs originating from random noise coincidences. This means that the search returns a coincident event that is caused by a particular noise realization which happens to coincide with an injection with an optimal SNR below the trigger threshold. Many of these injections should thus not be recoverable but are detected at high FAR due to these noise fluctuations. At a FAR of 10^3 per month we expect less then $\mathcal{O}(10)$ of these false associations. Another reason to only compare the sensitivity at low FARs of the machine learning and the matched filtering based searches are the thresholds used to find triggers. The matched filter search uses a threshold of SNR 5 whereas the machine learning search uses a threshold on the USR ranking statistic of -2.2. Because there is no direct relation between these two statistics, we cannot guarantee that both thresholds correspond to similar signal strengths. It may be possible that one search excludes weak signals which are found by the other based on this difference in the threshold.

The sensitivity difference between machine learning and matched filtering stays constant between using data from a single detector and using data from two detectors when matched filtering may only check for time consistency between detection candidates from the two observatories. The performance difference increases when matched filtering also checks for signal consistency. It is, therefore, reasonable to believe that a multi detector machine learning search may be more sensitive when it too can check for signal consistency. This would either require the single detector network to output parame-

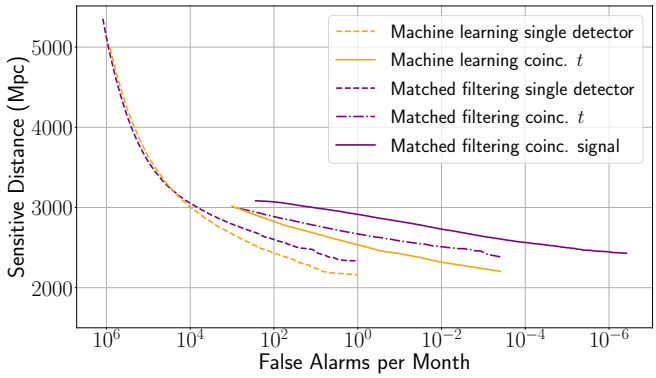

FIG. 2. Shown are the sensitive distances of different search algorithms as a function of the FAR. In orange we show the sensitivity curves of the machine learning based searches presented in [37] and this work. In purple we show sensitivity curves of an equivalent matched filter search. The dashed lines are derived on data only from a single detector. A label "coinc. t" refers to events being tested for coincidence based solely on the time difference of the events in the two detectors. The label "coinc. signal" means that the matched filter search also checked for signal consistency based on the time-, phase-, amplitude-difference, and intrinsic parameters in the two detectors. Sensitivities derived on data from more than one detector are truncated at a FAR of 10^3 per month due to an increasing number of true detections caused by random coincident events in the noise.

ter estimates of the detected signal alongside a ranking statistic or a single network that uses the data from both detectors as input. In the following section III we explore the second hypothesis.

III. TWO DETECTOR NETWORK

The deep learning algorithm presented in section II is significantly less sensitive than the full matched filter analysis that takes signal consistency into account. On the other hand, when the deep learning algorithm is compared to the matched filter search where signal consistency is ignored, the difference in sensitivity is comparable to the difference in sensitivity for a single detector. This gives reason to believe that the difference in sensitivity compared to the full matched filter search could be reduced when the network may operate on the data from both detectors and consider coincidences itself.

A. Architecture

We construct a network that uses data from both detectors while still retaining the ability to efficiently estimate a large background. The network from section II is still applied to the data from the two detectors individually. However, the final layer is removed and the 64 output-neurons from both networks are concatenated.

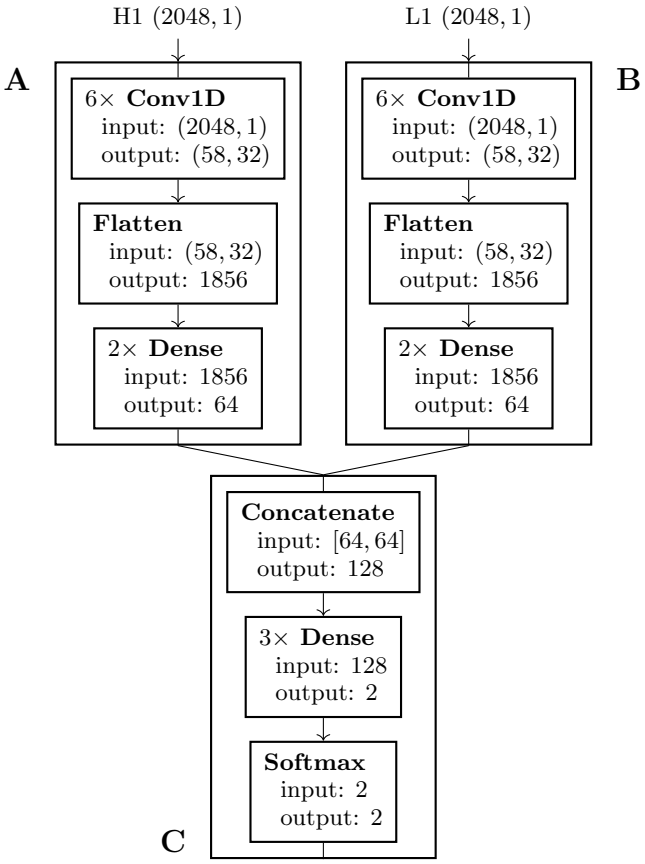

FIG. 3. A high level overview of the two-detector architecture. The network consists of three sub-networks A, B, and C. A detailed description of the sub-networks A and B can be found in Table I by removing the final row. The fully connected Dense layers contain 128, 64, and 2 neurons in that order. All but the final Dense layer are equipped with an exponential linear unit (ELU) activation.

We then add 3 more fully connected layers to look for coincidences between the detectors. An overview of the network is shown in Figure 3.

The last layer from the single detector network is removed to create a large latent space. A matched filter search compresses the input data into the ranking statistic, the time of the merger, and the parameters of the best matching template. The intention is that 64 neurons may be sufficient for a comparable compression and that the additional layers that operate on the concatenated outputs could perform a signal consistency analysis.

The sub-networks A and B in Figure 3 are intended to act as encoders that reduce the 2048 dimensional input into a latent space of dimension 64. It may be interesting in the future to train these sub-networks initially as autoencoders [48] from which only the encoder is used for detection purposes afterwards. Autoencoders are neural networks which in the most simple form consist of an encoder network and a decoder network. The encoder network compresses the input to some lower di-

mensional latent representation whereas the decoder uses that lower dimensional representation to reconstruct the input. Other studies have already found that autoencoders have potential applications in GW data analysis [49, 50].

B. Data Sets and Training

The network is trained on data similar to that presented in section II B. However, the data is extended to two detectors and sources are uniformly distributed in the sky. The latter change is required due to the amplitude and phase correlations in the two detectors.

We utilize the pre-trained single detector network used in section II in two different ways. In both cases the single detector parts of the two detector network (A and B in Figure 3) are initialized with the weights of the pre-trained model from section II. However, for one of the two networks, these weights are then not optimized during training, leaving only the weights of the final fully connected layers (C in Figure 3) to be adjusted. This approach is known as transfer learning [51] and has been successfully applied for different problems [52–54]. The second network optimizes the weights of the entire network. We also train a third network of the same architecture, where all parameters are initialized randomly and optimized during training.

The same optimizer settings and loss function described in section II B are used to train all three networks for 300 epochs. They are trained with a Softmax activation on the final layer, which is removed during evaluation. Each network is only trained once and the epoch with the highest efficiency on the validation set is chosen for further analysis.

C. Coincident Events

Because the networks output a single value when given the data from two detectors, we interpret that output as a coincidence ranking statistic at the corresponding time. We then perform the same clustering and thresholding described in section II C to obtain a list of coincident events.

D. Background Estimation

Determining the background of the two detector network is more challenging than for the single detector network from section II E, as there is no direct way of performing time shift in a computationally efficient way. One would, therefore, naively be limited by the duration of the analyzed data or would have to re-evaluate the entire month of test data multiple times. However, the network is designed in such a way that the data from both detectors are still analyzed individually and combined only at later stages. We evaluate the single detector data individually with the sub-networks A and B from Figure 3 and store those outputs. We then permute the order of the outputs from sub-network B such that it corresponds to a time shift with respect to the output from sub-network A. Finally, sub-network C is applied to the concatenated data from sub-network A and B for many different time shifts. Since sub-network C is very simple and time shifts can be generated trivially this process generates $\mathcal{O}(1000)$ months of background within $<$ 12 h on a NVIDIA RTX 2070 Super.

E. Evaluation and Comparison to Matched Filtering

Figure 4 shows the sensitive distance of the various networks as a function of the FAR and compares them to the results presented in section II H. All curves are truncated at a FAR of 10^3 per month due to the large number of false associations described in section II H. The three networks utilizing the data from both detectors described in this section are labeled as "Machine learning network coinc.". The matched filter results are shown in purple, where the dash-dotted line considers only time coincidence and the filled line also takes the consistency of intrinsic source parameters, phase, and amplitude into account. The orange line corresponds to the network from section II.

The networks described in this section were designed to be able to take signal consistency into account by reducing the input data to a large latent space. As such we were expecting sensitivities at low FARs to be larger than those obtained from time coincidence between single detector events produced by the single detector network.

However, we find that at low FARs all of the two detector networks are roughly as sensitive as the network tested in section II H. Therefore, they are still less sensitive than the matched filter equivalent and do not seem to take signal consistency into account. For high FARs, on the other hand, they are more sensitive. We suspect that the large time variation of the peak amplitude of ± 0.1 s may be responsible for this behavior. The networks are, thereby, trained to be insensitive to variations in timing of less then 0.1 s, which may produce phase and amplitude variations in a broad range.

IV. CONCLUSIONS

In this paper we have extended the single detector deep learning GW search algorithm from [20, 37] to two detectors and compared it to an equivalent matched filter algorithm. We found that the most simple extension, applying the one detector network to the data from two detectors individually and searching for coincident events, retains $\approx 92\%$ of the sensitivity of matched filter-

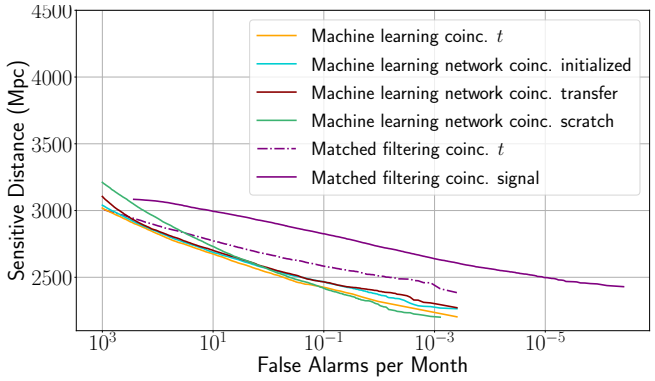

FIG. 4. The sensitivity of different search algorithms as a function of the FAR. All shown algorithms operate on the data from two detectors. The curves labeled "Machine learning coinc." are neural network search algorithms that consider data from both detectors and an overview can be found in Figure 3. The network labeled "initialized" initializes the sub-networks A and B as shown in Figure 3 from the single detector network used in section II H but optimizes them during the subsequent training. The network labeled "transfer" also initializes both sub-networks as the "initialized" network but freezes their weights. The network labeled "scratch" initializes all parameters of the network randomly. All other searches operate on the data from the individual detectors first and then search for coincident events. A label "coinc. t" refers to events being tested for coincidence based solely on the time difference of the events in the two detectors. The label "coinc. signal" means that the matched filter search also checked for signal consistency based on intrinsic parameters and the time-, phase-, and amplitude-difference in the two detectors. The curve labeled "Machine learning coinc. t" refers to the two-detector machine learning search analyzed in section II H. All sensitivities are truncated at a FAR of 10^3 per month due to a growing number of true positive detections caused by the coincidence of noise events.

ing, when only the time consistency between detectors is required. This fraction drops to $\approx 84\%$ when signal consistency between detectors is also considered.

To operate on data from two observatories, we constructed a two detector ranking statistic for the machine learning search based on the single detector USR ranking statistic proposed in [37]. This ranking statistic proved to be correlated with the network SNR.

We also highlighted the advantages of using a single detector network to construct a two detector search. Firstly, the single detector network does not need to be re-trained to be applied to the second detector, if both have similar noise characteristics. Secondly, this approach enables an efficient background estimation by applying relative time shifts to the recovered single detector events. This allows to test the two detector search to almost arbitrarily low FARs at low computational expenses. This method has already proven to be effective and reliable in state-of-the-art classical search algorithms [5, 15, 16].

Because using a single detector network restricts one to check for coincidences based solely on the timing difference, we tested a simple network that operates on data from both detectors directly. This allows the network in principle to construct internal signal representations which can be correlated between observatories. The network was constructed by removing the final layer of the single detector network, concatenating the outputs and adding a few fully connected layers to check for coincident events. The final fully connected layers, thus, receive 64 latent variables for each detector that can be checked for coincidence.

This design of the two detector network allowed us to do efficient background estimation. By applying relative time shifts to the outputs of the individual detector sub-networks, only the final few fully connected layers need to be evaluated for all shifts. The bulk of the computation, namely evaluating the input data of the detectors, only needs to be done once.

The network architecture was trained in three different ways; randomly initialized parameters for the entire network, parameters of the sub-networks initialized from of the single detector network, and parameters of the individual detector sub-networks fixed to the single detector parameters and optimizing only the final fully connected layers.

We found that all of these networks have very similar performance at low FARs. Neither of them performed substantially better than the initial network that looked for time coincident events between the single detector network outputs. It, therefore, seems as if the network architecture explored here is unable to learn any additional information about the signal. This may be caused by the allowed time-variance of $\pm 0.1\,\mathrm{s}$ for signals in the training set, which may limit the time resolution of the network and thus overshadow correlations in any other parameters. More sophisticated network architectures with higher time resolution may improve our findings. First promising steps have already been taken by [24]. Using an autoencoder to find a more meaningful latent representation of the input data may also be of use.

While the sensitivity was not improved by using a single network to process the data of two detectors, we still want to highlight that the method of determining the background may be of use for future networks.

Here we limited our research to GWs from non-spinning binary black holes with signal duration $< 1\,\mathrm{s}$ and Gaussian noise. Any of these simplifications are desirable to be lifted. Especially considering real noise may increase the gap in sensitivity between the single detector and multi detector search algorithm, by vetoing glitches. While we considered only two detectors an extension to a larger network should be trivial and may follow studies such as [55].

V. ACKNOWLEDGEMENTS

We thank Ondřej Zelenka, Frank Ohme, and Bernd Brügmann for valuable discussions and their scientific input. We acknowledge the Max Planck Gesellschaft and the Atlas cluster computing team at Albert-Einstein Institut (AEI) Hannover for support.

[1] J. Aasi *et al.* (LIGO Scientific Collaboration), Class. Quantum Grav. **32**, 074001 (2015), arXiv:1411.4547 [gr-qc].
[2] F. Acernese *et al.* (VIRGO), Class. Quantum Grav. **32**, 024001 (2015), arXiv:1408.3978 [gr-qc].
[3] T. Akutsu *et al.* (KAGRA), Nature Astron. **3**, 35 (2019), arXiv:1811.08079 [gr-qc].
[4] B. P. Abbott *et al.* (LIGO Scientific, Virgo), Phys. Rev. X **9**, 031040 (2019), arXiv:1811.12907 [astro-ph.HE].
[5] R. Abbott *et al.* (LIGO Scientific, Virgo), (2020), arXiv:2010.14527 [gr-qc].
[6] A. H. Nitz, C. D. Capano, S. Kumar, Y.-F. Wang, S. Kastha, M. Schäfer, R. Dhurkunde, and M. Cabero, (2021), arXiv:2105.09151 [astro-ph.HE].
[7] R. Abbott *et al.* (LIGO Scientific, KAGRA, VIRGO), Astrophys. J. Lett. **915**, L5 (2021), arXiv:2106.15163 [astro-ph.HE].
[8] C. Messick *et al.*, Phys. Rev. D **95**, 042001 (2017), arXiv:1604.04324 [astro-ph.IM].
[9] T. Dal Canton, A. H. Nitz, B. Gadre, G. S. Davies, V. Villa-Ortega, T. Dent, I. Harry, and L. Xiao, (2020), arXiv:2008.07494 [astro-ph.HE].
[10] T. Adams, D. Buskulic, V. Germain, G. M. Guidi, F. Marion, M. Montani, B. Mours, F. Piergiovanni, and G. Wang, Class. Quant. Grav. **33**, 175012 (2016), arXiv:1512.02864 [gr-qc].
[11] B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. E. Creighton, Phys. Rev. D **85**, 122006 (2012), arXiv:gr-qc/0509116.
[12] L. Nuttall *et al.*, Class. Quant. Grav. **32**, 245005 (2015), arXiv:1508.07316 [gr-qc].
[13] B. P. Abbott *et al.* (LIGO Scientific, Virgo), Phys. Rev. Lett. **116**, 131103 (2016), arXiv:1602.03838 [gr-qc].
[14] B. P. Abbott *et al.* (LIGO Scientific, Virgo), Class. Quant. Grav. **37**, 055002 (2020), arXiv:1908.11170 [gr-qc].
[15] S. A. Usman *et al.*, Class. Quant. Grav. **33**, 215004 (2016), arXiv:1508.02357 [gr-qc].
[16] S. Sachdev *et al.*, (2019), arXiv:1901.08580 [gr-qc].
[17] S. Klimenko *et al.*, Phys. Rev. D **93**, 042004 (2016), arXiv:1511.05999 [gr-qc].
[18] D. George and E. A. Huerta, Phys. Rev. D **97**, 044039 (2018), arXiv:1701.00008 [astro-ph.IM].
[19] D. George and E. A. Huerta, Phys. Lett. B **778**, 64 (2018), arXiv:1711.03121 [gr-qc].
[20] H. Gabbard, M. Williams, F. Hayes, and C. Messenger, Phys. Rev. Lett. **120**, 141103 (2018), arXiv:1712.06041 [astro-ph.IM].
[21] C. Dreissigacker and R. Prix, Phys. Rev. D **102**, 022005 (2020), arXiv:2005.04140 [gr-qc].
[22] M. B. Schäfer, F. Ohme, and A. H. Nitz, Phys. Rev. D **102**, 063015 (2020), arXiv:2006.01509 [astro-ph.HE].
[23] P. G. Krastev, K. Gill, V. A. Villar, and E. Berger, Phys. Lett. B **815**, 136161 (2021), arXiv:2012.13101 [astro-ph.IM].
[24] W. Wei, A. Khan, E. A. Huerta, X. Huang, and M. Tian, Phys. Lett. B **812**, 136029 (2021), arXiv:2010.15845 [gr-qc].
[25] W. Wei and E. A. Huerta, Phys. Lett. B **816**, 136185 (2021), arXiv:2010.09751 [gr-qc].
[26] E. Cuoco *et al.*, Mach. Learn. Sci. Tech. **2**, 011002 (2021), arXiv:2005.03745 [astro-ph.HE].
[27] E. A. Huerta and Z. Zhao, (2021), arXiv:2105.06479 [astro-ph.IM].
[28] W. Wei, E. A. Huerta, M. Yun, N. Loutrel, R. Haas, and V. Kindratenko, (2020), arXiv:2012.03963 [gr-qc].
[29] I. Harry, J. Calderón Bustillo, and A. Nitz, Phys. Rev. D **97**, 023004 (2018), arXiv:1709.09181 [gr-qc].
[30] I. Harry, S. Privitera, A. Bohé, and A. Buonanno, Phys. Rev. D **94**, 024012 (2016), arXiv:1603.02444 [gr-qc].
[31] A. H. Nitz, A. Lenon, and D. A. Brown, Astrophys. J. **890**, 1 (2019), arXiv:1912.05464 [astro-ph.HE].
[32] A. K. Lenon, D. A. Brown, and A. H. Nitz, (2021), arXiv:2103.14088 [astro-ph.HE].
[33] A. H. Nitz and Y.-F. Wang, (2021), 10.3847/1538-4357/ac01d9, arXiv:2102.00868 [astro-ph.HE].
[34] A. H. Nitz and Y.-F. Wang, (2021), arXiv:2106.08979 [astro-ph.HE].
[35] M. Zevin *et al.*, Class. Quant. Grav. **34**, 064003 (2017), arXiv:1611.04596 [gr-qc].
[36] R. Essick, P. Godwin, C. Hanna, L. Blackburn, and E. Katsavounidis, (2020), arXiv:2005.12761 [astro-ph.IM].
[37] M. B. Schäfer, O. Zelenka, A. H. Nitz, F. Ohme, and B. Brügmann, (2021), arXiv:2106.03741 [astro-ph.IM].
[38] A. Nitz, I. Harry, D. Brown, C. M. Biwer, J. Willis, T. D. Canton, C. Capano, L. Pekowsky, T. Dent, A. R. Williamson, G. S. Davies, S. De, M. Cabero, B. Machenschalk, P. Kumar, S. Reyes, D. Macleod, dfinstad, F. Pannarale, T. Massinger, S. Kumar, M. Tápai, L. Singer, S. Khan, S. Fairhurst, A. Nielsen, S. Singh, shasvath, and B. U. V. Gadre, "gwastro/pycbc: 1.18.0 release of pycbc," (2021).
[39] A. H. Nitz, T. Dent, T. Dal Canton, S. Fairhurst, and D. A. Brown, Astrophys. J. **849**, 118 (2017), arXiv:1705.01513 [gr-qc].
[40] C. Devine, Z. B. Etienne, and S. T. McWilliams, Class. Quant. Grav. **33**, 125025 (2016), arXiv:1601.03393 [astro-ph.HE].
[41] A. Bohé *et al.*, Phys. Rev. D **95**, 044028 (2017), arXiv:1611.03703 [gr-qc].
[42] L. S. Collaboration, "LIGO Algorithm Library - LALSuite," free software (GPL) (2018).

[43] D. P. Kingma and J. Ba, arXiv e-prints , arXiv:1412.6980 (2014), arXiv:1412.6980 [cs.LG].

[44] F. Chollet *et al.*, "Keras," `https://keras.io` (2015).

[45] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," (2015), software available from tensorflow.org.

[46] S. Husa, S. Khan, M. Hannam, M. Pürrer, F. Ohme, X. Jiménez Forteza, and A. Bohé, Phys. Rev. D **93**, 044006 (2016), arXiv:1508.07250 [gr-qc].

[47] S. Khan, S. Husa, M. Hannam, F. Ohme, M. Pürrer, X. Jiménez Forteza, and A. Bohé, Phys. Rev. D **93**, 044007 (2016), arXiv:1508.07253 [gr-qc].

[48] M. A. Kramer, AIChE Journal **37**, 233 (1991), https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/aic.690370209.

[49] H. Shen, D. George, E. A. Huerta, and Z. Zhao, (2019), 10.1109/ICASSP.2019.8683061, arXiv:1903.03105 [astro-ph.CO].

[50] H. Gabbard, C. Messenger, I. S. Heng, F. Tonolini, and R. Murray-Smith, (2019), arXiv:1909.06296 [astro-ph.IM].

[51] K. Weiss, T. M. Khoshgoftaar, and D. Wang, Journal of Big Data **3**, 9 (2016).

[52] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, in *Artificial Neural Networks and Machine Learning – ICANN 2018*, edited by V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis (Springer International Publishing, Cham, 2018) pp. 270–279.

[53] D. George, H. Shen, and E. A. Huerta, Phys. Rev. D **97**, 101501 (2018).

[54] R. Mesuga and B. J. Bayanay, (2021), 10.13140/RG.2.2.16403.20000, arXiv:2107.01863 [gr-qc].

[55] G. S. Davies, T. Dent, M. Tápai, I. Harry, C. McIsaac, and A. H. Nitz, Phys. Rev. D **102**, 022004 (2020), arXiv:2002.08291 [astro-ph.HE].