Exploring Gravitational Wave Selection Effects with Artificial Neural Networks

Nathan Moore,[1,*]
Advisors: Geraint Pratten,[2,†] and Patricia Schmidt[2,‡]

[1] *Department of Physics and Astronomy, Texas A&M University-Commerce, Commerce, TX 75428, USA*
[2] *School of Physics and Astronomy and Institute for Gravitational Wave Astronomy, University of Birmingham, Birmingham, B15 2TT, UK*

(Dated: August 2, 2022)

We present a study into using neural networks to predict the detectability of binary black hole mergers in ground-based gravitational wave detector networks. The probability of detection plays a key role in characterising selection effects, i.e. Malmquist biases, and is an important component in determining the underlying astrophysical population. In our analysis, we focused on three future detector networks and built a binary classifier for each scenario. The detector networks correspond to: i) the fourth observing run (O4, early 2023) with Hanford (H), Livingston (L) and Virgo (V), ii) the fifth observing run (O5, 2025) at A+ design sensitivity using the HLV network, and iii) O5 but including the Kagra (K) detector, HLVK. By construction, the networks map the intrinsic and extrinsic parameters (i.e. mass of the binary, mass ratio, redshift) of the binary to an output binary classification of *detectable* or *not detectable*. The networks allow us to incorporate additional physics that is typically missing in semi-analytical approximations. In particular, we use a state-of-the-art waveform model, IMRPhenomXPHM, that models precession and higher-order multipoles to produce the theoretical binary black hole signals. We briefly comment on alternative approaches, such as a direct regression network for the signal-to-noise ratio, and highlight some of the difficulties in constructing accurate networks.

I. INTRODUCTION

Gravitational wave (GW) observations provide a unique means to study the origin and evolution of astrophysical black holes and neutron stars. To date, we have seen on the order of 100 compact binaries, consisting of pairs of neutron stars or black holes [1]. However, it is important to introduce a metric that describes how *detectable* or *loud* GW signals are. Collecting data on how loud the theoretical signals are can help with large population studies, and can give an idea of what to use as selection effects during the observing runs. This detectability is typically translated into a probability by using the signal-to-noise ratio (SNR). The SNR is calculated by calculating the inner product of a GW signal weighted by the power spectral density (PSD) of the detector. Calculating the SNR, given intrinsic (the total mass of the binary, the mass ratio, the spins of each object) and extrinsic (sky location, polarization, and inclination angle describing how the binary is oriented relative to the observer) parameters of the binary is a straightforward and efficient calculation. Nonetheless it can still be computationally expensive to calculate the vast number of SNRs required to predict selection effects for the GW detectors.

Basic neural networks are constructed based on simple linear equations (neurons) that, when trained properly, can predict complex mathematical operations extremely efficiently. Using a neural network to predict the observability of GW signals proves to be a useful and fast tool in determining selection effects for future GW observing runs. In this analysis, we were particularly interested in the observability of precessing binaries and the impact of higher order multipoles.

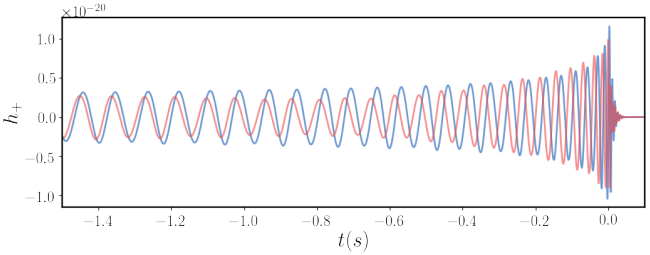

Figure 1. Two GW waveforms in the time-domain with the same total mass and mass ratio, one of which has spins aligned with its orbital angular momentum (blue) and the other which does not and therefore is precessing (red).

Precession occurs when spins are misaligned with the orbital angular momentum, resulting in the orientation of the orbital plane of the binary evolving with time [2]. This happens due to relativistic spin-orbit couplings. If the spins are preferentially (anti-)aligned with the orbital angular momentum, then the system will not precess and the orbital angular momentum direction will be fixed throughout the inspiral, merger and ringdown of the binary. A precessing GW signal has a number of unique characteristic features, such as modulating the maxima and minima of the amplitude as it evolves towards merger, shown in Fig. 1.

Higher order multipoles is a concept that comes from the spin-weighted spherical harmonic decomposition of the GW signal. The earliest calculations focused on the

* nmoore19@leomail.tamuc.edu
† G.Pratten@bham.ac.uk
‡ P.Schmidt@bham.ac.uk

dominant $(\ell, m) = (2, 2)$ mode, called the quadrupole. This is due to the fact that the quadrupole is the leading order mode of the signal at which we observe gravitational radiation [3]. However, higher multipoles will be excited if there is significant mass or spin asymmetry in the binary, meaning that they can play an important role in calculating the detectability in future GW detector networks. Shown in Fig. 2 are the other higher multipoles of a GW signal as compared to the full signal produce by all multipoles.

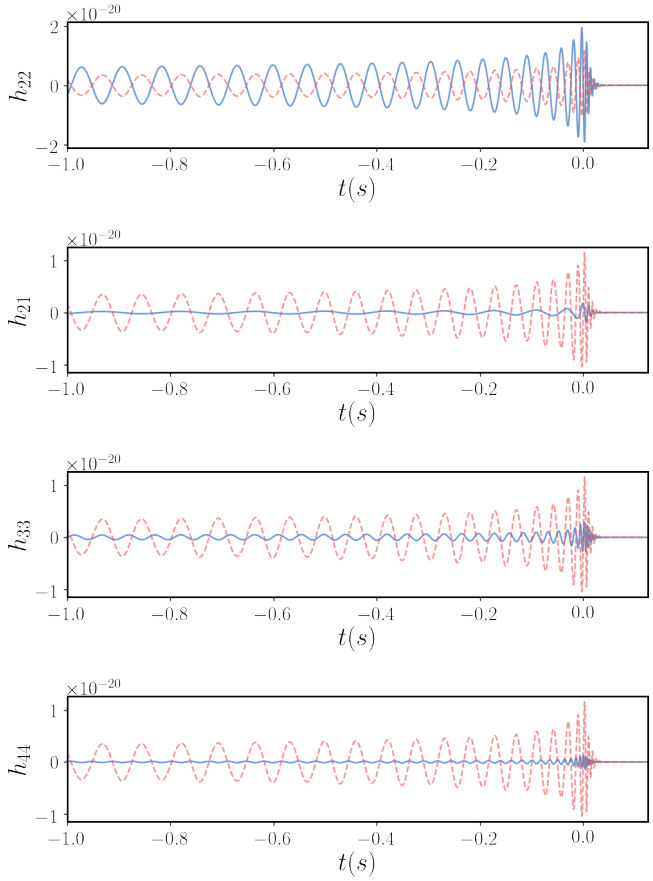

Figure 2. The time-domain waveform of each mode of a GW signal (blue) in comparison its entire waveform (red).

The likelihood that a signal can be received by a detector also depends on a minimum SNR intrinsic to the detector, so if this detector minimum is 12, it will only detect signals that have an SNR greater than 12. Using neural networks to determine whether or not a signal is detectable instead of explicitly calculating the SNR could prove to be more efficient [4].

II. METHODS

A. Gravitational Wave Data Analysis

A physical quantity called the strain h is the key observable in GW data analysis. The strain h can be written in terms of a spin-weighted spherical harmonic decomposition [5]

$$h = h_+ - ih_\times = \sum_{\ell \geq 2} \sum_{m=-\ell}^{\ell} h_{\ell m}\, {}_{-2}Y_{\ell,m}(\vartheta, \phi), \tag{1}$$

where $\{\vartheta, \phi\}$ are the sky location of the source in polar coordinates. Given an incoming GW with polarizations h_+ and $h_\times$, the real-valued response of a GW detector can be expressed as

$$h(t) = F_+ h_+(t) + F_\times h_\times(t), \tag{2}$$

where $\{F_+, F_\times\}$ denote the antenna response patterns of the detector [6]

$$F_+ = \frac{1}{2} \cos 2\psi \left(1 + \cos^2 \vartheta\right) \cos 2\varphi - \sin 2\psi \cos \vartheta \sin 2\varphi, \tag{3}$$

$$F_\times = \frac{1}{2} \sin 2\psi \left(1 + \cos^2 \vartheta\right) \cos 2\varphi + \cos 2\psi \cos \vartheta \sin 2\varphi, \tag{4}$$

where ψ is the gravitational-wave polarization. In Fig. 1, we show the real component of h, i.e. h_+, in the time-domain for a prototypical binary black hole. A generic precessing binary black hole with component masses m_1 and m_2 is governed by 7 intrinsic parameters: the mass ratio $q = m_2/m_1 \leq 1$ and the dimensionless spin vectors for each black hole $\boldsymbol{\chi}_i$. Here $\boldsymbol{\chi}_i = S_i/m_i^2$ where S_i is the physical spin of the black hole. The total mass $M = m_1 + m_2$ acts as an overall scaling parameter that shifts the GW frequencies to lower (higher) frequencies for heavier (lighter) masses. The position and orientation of the binary are specified with respect to a GW detector in terms of the right ascension $\alpha = \varphi$, declination $\delta = \pi/2 - \vartheta$, inclination ι, and polarization angle ψ. We combine these parameters into intrinsic and extrinsic parameters denoted by $\theta = \{M, q, \boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$ and $\lambda = \{\alpha, \delta, \iota, \psi\}$ respectively.

An important quantity in GW data analysis is the noise-weighted inner product, or the overlap, defined by

$$\langle h_1, h_2 \rangle = 2 \int_0^\infty \frac{df}{S_n(f)} \left[\tilde{h}_1(f) \tilde{h}_2^*(f) + \tilde{h}_1^*(f) \tilde{h}_2(f)\right], \tag{5}$$

which defines the level of agreement or disagreement between two waveforms h_1 and h_2. The match is defined as

$$\mathcal{M}(h_1, h_2) = \max_{t_0, \varphi_0} \langle h_1, h_2 \rangle, \tag{6}$$

where $\{t_0, \varphi_0\}$ are the coalesence time and coalesence phase respectively. Similarly, the signal-to-noise ratio (SNR) can be defined by

$$\rho^2 = \langle h_1, h_1 \rangle = 4 \int_0^\infty \frac{|\tilde{h}_1(f)\tilde{h}_1^*(f)|}{S_n(f)} df, \tag{7}$$

and quantifies how loud a GW signal is in a detector with a noise power spectral density $S_n(f)$.

B. Detection Probability

The SNR plays a key role in determining the selection effects for GW observations. In particular, the detectability of a binary can be defined as

$$p_{\rm det}(\theta, z) = \int p(\lambda) \Theta[\rho(\theta, z, \lambda) - \rho_{\rm thr}] d\lambda, \tag{8}$$

where $\rho(\theta, z, \lambda)$ is the SNR of the binary given intrinsic parameters θ and $\rho_{\rm thr}$ is the network threshold SNR that determines when a binary is detectable. Following [4, 7, 8], we use a conservative detection threshold for a network of GW detectors of $\rho_N > 12$. We note that thresholding detections based on the SNR does not fully account for the empirical distribution of triggers generated by the search pipelines [4, 9, 10] and that the SNR alone is a poor discriminator for determining the astrophysical probability of a binary [11].

The detection probability $p_{\rm det}$ allows us to define the sensitive spacetime volume VT which quantifies the volume over which a GW detector network is expected to detect coalescing compact binaries throughout an observing run

$$VT(\theta) = T_{\rm obs} \int p_{\rm det}(\theta, z) \frac{dV_c}{dz} \frac{1}{1+z} dz, \tag{9}$$

where $T_{\rm obs}$ is the total observing time and V_c is the comoving volume. This makes the detection probability $p_{\rm det}$ a key ingredient in inferring the population hyperparameters Λ that characterize the underlying astrophysical distribution of binaries. In particular, it is used to calculate the detection fraction [12–15]

$$\xi(\Lambda) = \int p_{\rm det}(\theta)\, \pi(\theta|\Lambda)\, d\theta, \tag{10}$$

where θ denotes the binary parameters. The number of expected detections over a GW observing run is

$$N_{\rm exp} = N\, \xi(\Lambda), \tag{11}$$

where N denotes the total number of expected events both resolvable and unresolvable. Accurately determining $p_{\rm det}$ will therefore be important in making unbiased inferences on the underlying astrophysical population, especially in the context of precession and higher multipoles [4].

The SNR in Eq. (8) has traditionally been calculated using a simplification originally introduced in [16]. In this approximation, the SNR is calculated by weighting the SNR of a optimally oriented source $\rho_{\rm opt}$, i.e. one that is face-on and overhead, by some angular projection factor ω. This approximation is valid when characterizing the response of a single GW detector to the angular distribution of power for a non-precessing binary using only the $(\ell, m) = (2, 2)$ multipole. This approximation can be derived by looking at the leading order GW emission from a compact binary

$$h_+(t) = A(t) \frac{1 + \cos^2 \iota}{2} \cos \varphi(t), \tag{12}$$

$$h_\times(t) = A(t) \cos \iota \sin \varphi(t), \tag{13}$$

where $A(t)$ is the GW amplitude and $\varphi(t)$ the phase. In particular, we simplify Eq. (1) as [4, 16]

$$h(t) = \omega A(t) \cos\left[\varphi(t) - \varphi_0\right], \tag{14}$$

where

$$\omega = \left[\left(F_+ \frac{1 + \cos^2 \iota}{2} \right)^2 + \left(F_\times \cos \iota \right)^2 \right]^{1/2}, \tag{15}$$

$$\tan \varphi_0 = \frac{2 F_\times \cos \iota}{F_+ \left(1 + \cos^2 \iota\right)}. \tag{16}$$

By construction, the angular projection factor has a maximum

$$\max_{\theta, \varphi, \iota, \psi} \omega = 1, \tag{17}$$

that occurs when the source is optimally oriented, i.e. face-on ($\iota = 0$) and overhead ($\theta = 0$). We can therefore factorize the SNR into the optimally oriented SNR $\rho_{\rm opt}$ and the angular projection factor ω as

$$\rho_{\rm S}(\theta, z, \lambda) = \omega(\lambda)\, \rho_{\rm S,opt}(\theta, z). \tag{18}$$

From Eq. (18), we can calculate $p_{\rm det}$ using standard Monte Carlo integration methods. In the single detector limit, we adopt an SNR threshold of $\rho_S > 8$ as opposed to the threshold $\rho_N > 12$ that we adopt for the network of detectors.

As noted above, however, this approach towards calculating the probability of detection for a GW signal relies on using an optimally oriented binary, is only based on the SNR predicted by a single detector and is restricted to non-precessing binaries. In reality, a typical GW detector network will typically have at least 2 or more detectors operational at any one time, due to the duty cycle of the detectors, and we expect astrophysical binaries to have spins that are both generically oriented, i.e. precessing, and include higher multipoles beyond the $(2, 2)$-mode.

Evaluating the SNR at a single point in the parameter space for a generically precessing binaries can computationally efficient but calculating the SNR over an entire

population of binaries will be computationally expensive. This all the more problematic when one has to appeal to Monte Carlo methods to estimate $p_{\rm det}$, as per Eq. (8). A possible alternative strategy, as explored in this paper, is to construct a neural network to estimate the detectability of a binary [4]. In this approach, we pre-calculate the SNRs for a population of binaries and determine which of these binaries are detectable ($\rho \geq \rho_{\rm thr}$) and which are not, effectively reducing the problem to a binary classifier. We therefore train a neural network binary classifier that effectively estimates the term $\Theta\left[\rho\left(\theta, z\lambda\right) - \rho_{\rm thr}\right]$ appearing in Eq. (8). A particular advantage of this approach is that we can train a network using theoretical models for the observed GW signal that incorporate all relevant physics, notably precession and higher multipoles. This circumvents the limitation to the non-precessing 22-only waveform models as used in Eq. (18). A caveat, however, is that we must pre-specify the detection threshold before training the network. If we change $\rho_{\rm thr}$, we must train a new neural network, though we can recycle the pre-computed SNRs. As detailed above, we assume a conservative detection threshold of $\rho_N > 12$ in all neural networks presented in this paper.

In order to create the training and validation datasets used by the neural networks, we used a state-of-the-art waveform model, IMRPhenomXPHM [17–19], that includes both precession and higher multipoles. The priors used to construct the binary population assume a uniform prior on the total mass $M \in [2, 500]$, uniform mass ratios $q \in [0.05, 1.0]$ and a uniform distribution for the redshift $z \in \left[5 \times 10^{-3}, 2.0\right]$. The spin vectors for each black hole were taken to be isotropic on the sphere and uniform in spin magnitude, $\chi_i \in [0, 1]$. We used an isotropic prior for the location of the source on the sky and a prior on the inclination that was uniform in $\cos\iota$. The priors for the polarization ψ and coalesence phase φ were also taken to be uniform. Training data used for these networks included sets on the order of 10^7 binaries, with validation sets on the order of 10^5 binaries.

The SNR, as defined in Eq. (7), depends on the PSDs through $S_n(f)$. As such, we must construct training and validation data for each detector network considered in this analysis. We consider three detector networks that represent probable configurations that will be operating in the near future. The first network is the 3-detector HLV O4 network, due to begin operation in early 2023. The second network is a standard HLV detector network at projected O5 sensitivities, due to begin operation $\sim$ 2025. The final detector network is also and O5 network but now incorporating the KAGRA detector, HLVK. The PSDs for all detectors considered in this analysis are shown in Fig. 3.

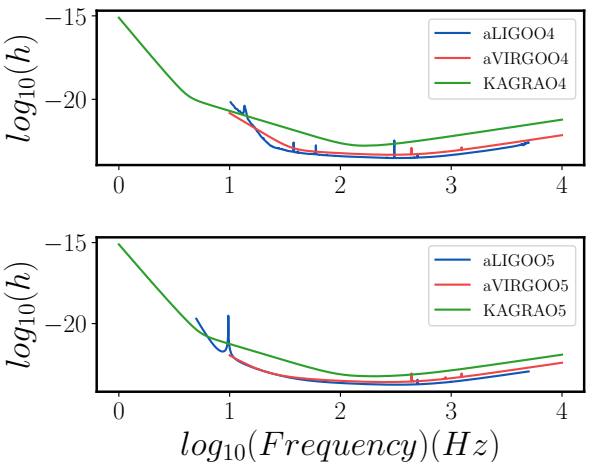

Figure 3. The PSDs used in the O4 (top) and O5 (bottom) networks. In the O4 neural network, the SNR was calculated from all three PSDs shown, as it was simulating the HLVK detector network. In the O5HLV neural network, the KAGRA PSD was omitted from the calculation, but it is used in the O5HLVK neural network.

III. EXPLORING NEURAL NETWORKS

A. Introduction to Machine Learning

In order to tackle the high-dimensionality of the parameter space, we construct a neural network binary classifier using `TensorFlow` [20] with a particular focus on the sequential model in `Keras` framework. In this section, we briefly review the fundamental operation of a neural network. A neural network is composed of a collection of connected nodes called neurons. The most basic network we could consider is a single neuron with a N inputs x_i and a single output y. Each input will be associated a particular weight w_i and the neuron itself can be given a bias b. In a prototypical *feedforward* network, the connections are directed from the input to the output of the neuron. The behaviour of a neuron is characterised by two operations: i) application of the weights and biases to the input data and ii) a (non-)linear transformation determined by the *activation function* σ

$$\hat{y}^{(i)} = \sigma\left(z^{(i)}\right) = \sigma\left(w^T x^{(i)} + b\right), \tag{19}$$

where a typical choice for the activation function is something like the sigmoid function

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \tag{20}$$

Here $\hat{y}$ is called the estimator of y. We can now introduce the *loss* function, which can be any metric that represents the error between the network estimated $\hat{y}$ and the true values y. For example, when performing regression the

loss is often taken to be the mean absolute error, the mean absolute percentage error or the mean squared error. In the binary case, however, an appropriate loss function to use is the binary cross entropy [21]

$$J = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(\hat{y}^{(i)}, y^{(i)}), \tag{21}$$

where $\mathcal{L}(\hat{y}^{(i)}, y^{(i)})$ is defined as

$$\mathcal{L}(\hat{y}^{(i)}, y^{(i)}) = -y^{(i)} \log(\hat{y}^{(i)}) - (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}). \tag{22}$$

Of particular importance is that the loss can be used to inform the network, i.e. make the network "learn". This is done using stochastic gradient descent algorithms, in which the parameters of the network are updated based on the derivatives of the loss in Eq. (23)

$$\theta = \theta - \alpha \frac{\partial J}{\partial \theta}, \tag{23}$$

where $\theta = \{w, b\}$, the weights and biases of the network, and α is the learning rate. Gradient descent is like trying to find the minima of a function by experimentally guessing locations along the function for the minima and using the derivative of the function at each point to determine the direction that each point should take to get closer to a minima. If this process was put to motion, we could treat the loss as a potential such that it would look like the points were rolling down the function and pooling in the minima of the potential, like stones rolling down a hill. In the neural network, the stones would be the network parameters θ and the network would be most accurate once it reaches the *global* minima of the loss. The learning rate α effectively determines how quickly the network adapts to the shape of the loss (or its derivatives). Typically the learning rate is set to a sufficiently small number, e.g. 10^{-2}, so that the weights and biases optimally approach the global minima, producing a more accurate and stable network. If the learning rate is too large, then updates to θ can be too large and step beyond or below the values needed for peak accuracy, converging to suboptimal minima. If, however, the learning rate is too small, the network can fail to reach the required accuracy as the weights and biases are evolving too slowly. This can be particularly problematic if the parameters become trapped in a suboptimal local minima.

In practice, we use more sophisticated algorithms within the `Keras` framework to dynamically update the learning rate. In particular, we use an adaptive scheme that reduces the learning rate when improvements to the loss have plateaud, this is provided by the `ReduceLROnPlateau` callback in `Keras`.

B. Optimizing Hyperparameters

To determine the final structure of the neural networks we optimized over the network hyperparameters. In particular we explored the role of the network architecture (i.e. how many layers and how many neurons in each layer), the standardization of the input data, the activation functions of each layer, batch size, learning rate, optimizer and the initializer used for the weights and biases.

As part of our hyperparameterization optimization, we explored different network architectures by varying the number of hidden layers and the number of neurons in each layer. In the O4-HLV network, the optimal configuration turned out to be 26 neurons in the first hidden layer, 52 neurons in the second layer, and 104 neurons in the final hidden layer, which we refer to as the *triangle* network. As part of our experimentation, we also tried a *basic* configuration $\{16, 16\}$, a *rhombus* network $\{26, 52, 104, 52, 26\}$, a *deep and narrow* network $\{13, 13, 13, 13, 13, 13\}$ and a *shallow and wide* network $\{104, 104\}$.

Standardization is a practice used for neural networks because large range values are difficult for them to work with. For example, in the training data for these networks that were put together for this project, the range of values for the total mass of the black hole binary was between 2 and 500 $M_\odot$. Using minimum-maximum scaling though, you can reduce the range to [-1,1] using the following

$$x' = -1 + 2\frac{x - \min(x)}{\max(x) - \min(x)}. \tag{24}$$

All of the data is still there, but with the values reduced to have these small differences the neurons can digest that information much easier. We also tried using normalization and reducing the range to [0,1].

Activation functions, like the sigmoid function, were explored for each layer of the final networks. The final networks only used the sigmoid activation function in the output layer, with all the hidden layers using a tanh activation function

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{25}$$

Because such large data sets are used in training neural networks it is common to use a batch size greater than one, meaning that rather than looking at a single input and a single output and determining the loss of that instance, the network can look at multiple sets of inputs and outputs and calculate the average loss and apply the changes based on the average. It can make the network take more epochs to become accurate, but it takes much less computing time to use larger batches. For debugging the neural networks a batch size of 1024 was used to make the network train faster. Once we were satisfied with all of the other hyperparameters we reduced it to 128, in order to increase accuracy at the expense of speed.

We decided on an initial global learning rate of 0.01, but included a callback function within the tensorflow libary that monitors whether or not the network is learning efficiently, and will change the learning rate as required.

Optimizers are the functions that carry out the gradient descent, which introduce some techniques to make the

	Training		Validation	
Network	Accuracy	Loss	Accuracy	Loss
O4HLV	0.981	0.042	0.968	0.073
O5HLV	0.972	0.063	0.919	0.062
O5HLVK	0.972	0.062	0.915	0.063

Table I. The accuracy of each neural network in correctly classifying the binaries in their Training data and their Validation data, as well as the final loss values.

training smoother. The Adagrad family of optimizers in TensorFlow use parameter-specific learning rates, that is a learning rate for each weight and bias, which is determined by how important each parameter is in achieving accurate output. The final networks used the Adadelta optimizer from this group which uses a window of network updates to optimize the learning rate rather than the accumulation of all previous updates as per Adagrad.

In the binary classifiers networks, we use the Glorot algorithm [22] to initialize the weights of the network, as implemented in `TensorFlow`. The Glorot algorithm randomly assigns values on a domain between $[-l,\ l]$ where $-l$ is calculated from $l = \sqrt{\frac{6}{i+o}}$ where i is the number of neurons used in determining the input of the neuron which is being initialized, and o is the number of neurons which directly use the output of the neuron being initialized. In the first hidden layer of the O4-HLV network this would be $l = \sqrt{\frac{6}{13+52}}$ because the neurons in this layer's inputs are the values from the input layer of 13 intrinsic parameters, and the outputs of this layer are directly used in the second hidden layer of 52 neurons.

IV. CALCULATING THE DETECTION PROBABILITY FOR O4 AND O5

The final neural networks used the same hyperparameters: the *triangle* structure, the Adadelta optimizer, binary cross entropy for its loss function, the Glorot uniform initializer, batch size of 128, and initial learning rate of 0.01. Table I shows the final accuracies of each of the networks. Calculating $p_{\rm det}$ with the O4HLV network in place of the Heaviside function of (8) produces results that would indicate preferred accuracy over the sub-optimal approximation in Eq. (18). To show how the neural networks are better at calculating $p_{\rm det}$ in the case of precessing waveforms, it is useful to acquire a metric for how much a binary system is precessing. For this we use χ_p which is calculated using the spin vectors for each object in the binary system [23]. The starting point is to approximately calculate the magnitude of the two in-plane spin contributions from each of the black holes in the system [23]

$$S_p := \frac{1}{2}(A_1 S_{1\perp} + A_2 S_{2\perp} + |A_2 S_{2\perp} - A_1 S_{1\perp}|), \quad (26)$$

where $A_1 = 2 + 3q/2$, $A_1 = 2 + 3/(2q)$ and $q \leq 1$. Using this average, [23] was able to construct a dimensionless effective precession parameter that captures the leading order effects of precession on the waveform

$$\chi_p := \frac{S_p}{A_1 m_1^2}. \quad (27)$$

In this analysis, we use χ_p as a measure of the amount of precession in the system. In Fig. 5 and Fig. 6, we show the difference in the estimated $p_{\rm det}$ between our neural network, incorporating precessing and higher multipoles, and the sub-optimal approximation based purely on a non-precessing waveform using only the 22-mode. We clearly see that $p_{\rm det}$ shows notable discrepancies across the entire precession spectrum with the strongest disagreement occurring at low mass ratios, where the mass asymmetry in the binary is largest.

V. TOWARDS SNR REGRESSION

A key limitation of the neural networks constructed here are that they are predicated on a somewhat ad-hoc choice of the threshold SNR. Recent alternative approaches that can help circumvent this issue include constructing a neural network regressor that estimates the SNR in a given detector network, e.g. [24]. Another approach, introduced in [25], uses density estimation on the set of detected binaries to produce a continuous, generative model for $p_{\rm det}$ valid for arbitrary subsets of the binary parameters. In this section, we briefly highlight initial work towards a neural network regressor and highlight some of the challenges in training a network in a high dimensional and highly degenerate parameter space. Attempting to create a neural network that performs this function means we had to use a loss function that was not the `BinaryCrossentropy` because the output needed to represent an real number rather than a category. Experimenting with loss function between `MeanSquaredError`, `MeanAbsoluteError`, and `MeanAbsolutePercentageError` proved that `MeanAbsoluteError` was the best, though poor results were still obtained. Also experimenting from the same group of network structures as before, *triangle* and *shallow and wide* often tied for least error. For optimizers we tried `Adagrad`, `Adam`, `Adamax`, and `Adadelta`, but settled on the last one as with the binary classifiers as the optimizer of least error. For standardization of data we ended on min-max scaling for the input parameters as with the binary classifier, but with normalizing the SNR intended output, see Eq. (28)

$$z = \frac{x - \mu}{\sigma}, \quad (28)$$

where μ is the mean SNR and σ is the standard deviation. With these hyperparameters in place, including batch

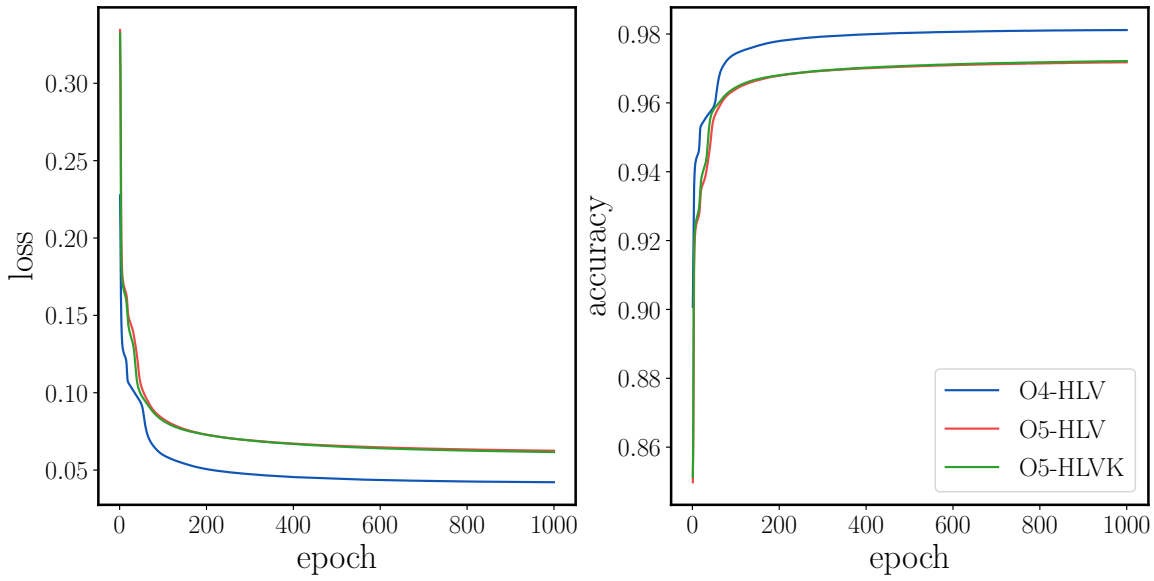

Figure 4. Loss and Accuracy plots for the three neural networks.

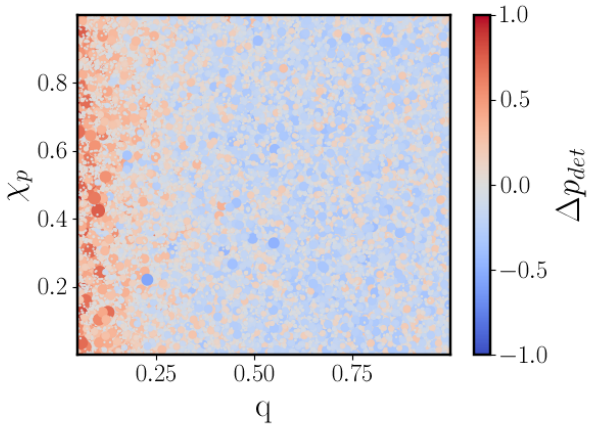

Figure 5. The difference in p_{det} between the binary classifier calculation and sub-optimal calculation for the binaries used in the training data for the O4-HLV neural network. Showing where the binaries lie in terms of mass ratio and χ_p.

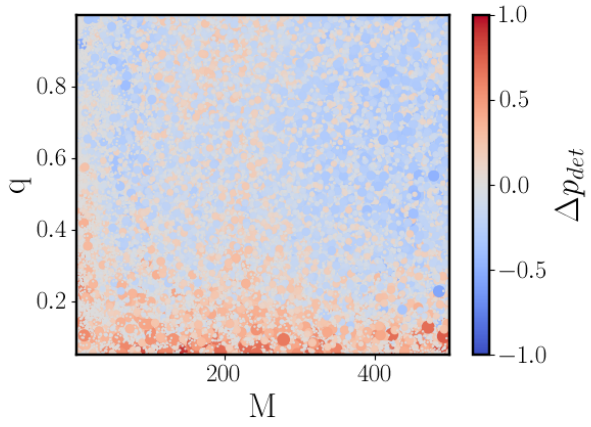

Figure 6. Same data as Fig. 5, showing how the binaries most impacted by the semi-analytical approximation are highly asymmetric ($q \to 0$) and typically at heavier masses, where higher multipoles are louder.

sizes of 1024 and initial global learning rate of 0.0001, and SNRs from the O5-HLV detector network, using the `linear` activation function for each hidden layer and the output layer, the resulting neural network had the final error of 156.39% and loss 0.004.

VI. CONCLUSION

We presented an approach that, while not necessarily novel in its entirety, introduces precession and higher-order modes to the problem of determining selection effects in detector networks with neural networks. Three neural networks were trained to high accuracy to simulate detector networks in future GW observing runs. Our results showed that probability of detection is higher than previously estimated, particularly in the dimension-

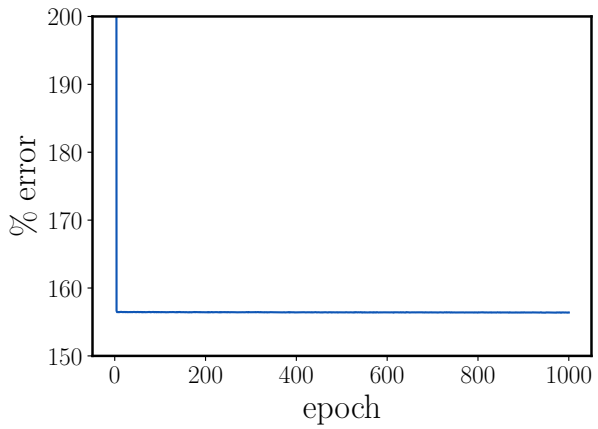

Figure 7. Attempted SNR regression network, displaying that it is not learning how to calculate SNR values.

space that would indicate excitation of higher-order modes and across all precession possibility. We hope that these findings are helpful in quickly and easily determining spacetime-volume estimates for binary systems with full-dimensionality, as well as more efficiently modeling the population of binaries that could be detected in future observing runs. The three successful networks can produce detectable/not detectable outputs given the 13 parameters, and while the concept of a neural network that can explicitly calculate SNRs was explored in this project, it certainly deserves more exploration. A successful attempt at such an approach could make the probability of detection calculations faster, among other applications. Another such approach that was not explored is a neural network that directly predicts the probability of detection, simulating Equation 8. A successful attempt at this approach would eliminate the need for producing the probability distribution of extrinsic parameters in the calculation, and thus heavily reduce computation requirements. Limitations on the neural networks presented in this paper include having to be trained to fit new power spectral densities for proposed detector networks as they become published, and lower detector thresholds as the technology improves. We recommend that these networks would be used in large injection campaigns, to estimate detection rates for future observing runs. We believe that neural networks represent an accurate and technologically efficient approach to GW data analysis.

ACKNOWLEDGMENTS

We thank Alf for useful discussions and comments. N.M. offers many thanks for the National Science Foundation's support from grant No. 1460803 and 1950830. G.P. and P.S. acknowledge support from STFC grant No. ST/V005677/1. Computations were performed on the Bondi HPC cluster at the Birmingham Institute for Gravitational Wave Astronomy. G.P. gratefully acknowledges support from an NVIDIA Academic Hardware Grant.

REFERENCES

[1] R. Abbott *et al.* (LIGO Scientific, VIRGO, KAGRA), (2021), arXiv:2111.03606 [gr-qc].
[2] T. A. Apostolatos, C. Cutler, G. J. Sussman, and K. S. Thorne, Phys. Rev. D **49**, 6274 (1994).
[3] A. Einstein, Sitzungsber. Preuss. Akad. Wiss. Berlin (Math. Phys.) **1918**, 154 (1918).
[4] D. Gerosa, G. Pratten, and A. Vecchio, Phys. Rev. D **102**, 103020 (2020), arXiv:2007.06585 [astro-ph.HE].
[5] K. S. Thorne, Rev. Mod. Phys. **52**, 299 (1980).
[6] B. S. Sathyaprakash and B. F. Schutz, Living Rev. Rel. **12**, 2 (2009), arXiv:0903.0338 [gr-qc].
[7] B. P. Abbott *et al.* (KAGRA, LIGO Scientific, Virgo, VIRGO), Living Rev. Rel. **21**, 3 (2018), arXiv:1304.0670 [gr-qc].
[8] B. P. Abbott *et al.* (LIGO Scientific, Virgo), Astrophys. J. Suppl. **227**, 14 (2016), arXiv:1606.03939 [astro-ph.HE].
[9] A. H. Nitz, T. Dent, T. Dal Canton, S. Fairhurst, and D. A. Brown, Astrophys. J. **849**, 118 (2017), arXiv:1705.01513 [gr-qc].
[10] C. Hanna *et al.*, Phys. Rev. D **101**, 022003 (2020), arXiv:1901.02227 [gr-qc].
[11] G. Pratten and A. Vecchio, Phys. Rev. D **104**, 124039 (2021), arXiv:2008.00509 [gr-qc].
[12] I. Mandel, W. M. Farr, and J. R. Gair, Mon. Not. Roy. Astron. Soc. **486**, 1086 (2019), arXiv:1809.02063 [physics.data-an].
[13] B. P. Abbott *et al.* (LIGO Scientific, Virgo), Astrophys. J. Lett. **882**, L24 (2019), arXiv:1811.12940 [astro-ph.HE].
[14] R. Abbott *et al.* (LIGO Scientific, Virgo), Astrophys. J. Lett. **913**, L7 (2021), arXiv:2010.14533 [astro-ph.HE].
[15] R. Abbott *et al.* (LIGO Scientific, VIRGO, KAGRA), (2021), arXiv:2111.03634 [astro-ph.HE].
[16] L. S. Finn and D. F. Chernoff, Phys. Rev. D **47**, 2198 (1993), arXiv:gr-qc/9301003.
[17] G. Pratten *et al.*, Phys. Rev. D **103**, 104056 (2021), arXiv:2004.06503 [gr-qc].
[18] C. García-Quirós, M. Colleoni, S. Husa, H. Estellés, G. Pratten, A. Ramos-Buades, M. Mateu-Lucena, and R. Jaume, Phys. Rev. D **102**, 064002 (2020), arXiv:2001.10914 [gr-qc].
[19] G. Pratten, S. Husa, C. Garcia-Quiros, M. Colleoni, A. Ramos-Buades, H. Estelles, and R. Jaume, Phys. Rev. D **102**, 064001 (2020), arXiv:2001.11412 [gr-qc].
[20] 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.

[21] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).

[22] X. Glorot and Y. Bengio, in *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (2010) pp. 249–256.

[23] P. Schmidt, F. Ohme, and M. Hannam, Phys. Rev. D **91**, 024043 (2015), arXiv:1408.1810 [gr-qc].

[24] K. W. K. Wong, K. K. Y. Ng, and E. Berti, (2020), arXiv:2007.10350 [astro-ph.HE].

[25] C. Talbot and E. Thrane, Astrophys. J. **927**, 76 (2022), arXiv:2012.01317 [gr-qc].