

PAPER • OPEN ACCESS

## Statistically-informed deep learning for gravitational wave parameter estimation

To cite this article: Hongyu Shen *et al* 2022 *Mach. Learn.: Sci. Technol.* **3** 015007

View the [article online](#) for updates and enhancements.

### You may also like

- [Automatic differentiation to simultaneously identify nonlinear dynamics and extract noise probability distributions from data](#)  
Kadierdan Kaheman, Steven L Brunton and J Nathan Kutz
- [InClass nets: independent classifier networks for nonparametric estimation of conditional independence mixture models and unsupervised classification](#)  
Konstantin T Matchev and Prasanth Shyamsundar
- [Inverse Dirichlet weighting enables reliable training of physics informed neural networks](#)  
Suryanarayana Maddu, Dominik Sturm, Christian L Müller et al.



## PAPER

## OPEN ACCESS

RECEIVED  
17 September 2021REVISED  
25 October 2021ACCEPTED FOR PUBLICATION  
10 November 2021PUBLISHED  
30 November 2021

Original Content from  
this work may be used  
under the terms of the  
[Creative Commons  
Attribution 4.0 licence](#).

Any further distribution  
of this work must  
maintain attribution to  
the author(s) and the title  
of the work, journal  
citation and DOI.



# Statistically-informed deep learning for gravitational wave parameter estimation

Hongyu Shen<sup>1,2,\*</sup> , E A Huerta<sup>3,4,\*</sup> , Eamonn O'Shea<sup>5</sup>, Prayush Kumar<sup>5,6</sup> and Zhizhen Zhao<sup>1,2,7,\*</sup> 

<sup>1</sup> Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, United States of America

<sup>2</sup> Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801, United States of America

<sup>3</sup> Data Science and Learning Division, Argonne National Laboratory, Lemont, IL 60439, United States of America

<sup>4</sup> Department of Computer Science, University of Chicago, Chicago, IL 60637, United States of America

<sup>5</sup> Cornell Center for Astrophysics and Planetary Science, Cornell University, Ithaca, NY 14853, United States of America

<sup>6</sup> International Centre for Theoretical Sciences, Tata Institute of Fundamental Research, Bangalore 560089, India

<sup>7</sup> National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, IL 61801, United States of America

\* Authors to whom any correspondence should be addressed.

E-mail: [hongyu2@illinois.edu](mailto:hongyu2@illinois.edu), [eliu@anl.gov](mailto:eliu@anl.gov) and [zhizhenz@illinois.edu](mailto:zhizhenz@illinois.edu)

**Keywords:** gravitational wave, deep learning, parameter estimation, posterior inference

## Abstract

We introduce deep learning models to estimate the masses of the binary components of black hole mergers,  $(m_1, m_2)$ , and three astrophysical properties of the post-merger compact remnant, namely, the final spin,  $a_f$ , and the frequency and damping time of the ringdown oscillations of the fundamental  $\ell = m = 2$  bar mode,  $(\omega_R, \omega_I)$ . Our neural networks combine a modified WaveNet architecture with contrastive learning and normalizing flow. We validate these models against a Gaussian conjugate prior family whose posterior distribution is described by a closed analytical expression. Upon confirming that our models produce statistically consistent results, we used them to estimate the astrophysical parameters  $(m_1, m_2, a_f, \omega_R, \omega_I)$  of five binary black holes: GW150914, GW170104, GW170814, GW190521 and GW190630. We use PyCBC Inference to directly compare traditional Bayesian methodologies for parameter estimation with our deep learning based posterior distributions. Our results show that our neural network models predict posterior distributions that encode physical correlations, and that our data-driven median results and 90% confidence intervals are similar to those produced with gravitational wave Bayesian analyses. This methodology requires a single V100 NVIDIA GPU to produce median values and posterior distributions within two milliseconds for each event. This neural network, and a tutorial for its use, are available at the Data and Learning Hub for Science.

## 1. Introduction

The advanced LIGO [1, 2] and advanced Virgo [3] observatories have reported the detection of tens of gravitational wave sources [4–6]. At design sensitivity, these instruments will be able to probe a larger volume of space, thereby increasing the detection rate of sources populating the gravitational wave spectrum. Thus, given the expected scale of gravitational wave discovery in upcoming observing runs, it is in order to explore the use of computationally efficient signal-processing algorithms for gravitational wave detection and parameter estimation.

The rationale to develop scalable and computationally efficient signal-processing tools is apparent. Advanced gravitational wave detectors will be just one of many large-scale science programs that will be competing for access to oversubscribed and finite computational resources [7–10]. Furthermore, transformational breakthroughs in multi-messenger astrophysics over the next decade will be enabled by combining observations in the gravitational, electromagnetic and astro-particle spectra. The combination of these high dimensional, large volume and high speed datasets in a timely and innovative manner presents unique challenges and opportunities [11–13].

The realization that companies such as Google, YouTube, among others, have addressed some of the big-data challenges we are facing in multi-messenger astrophysics has motivated a number of researchers to learn what these companies have done, and how such innovation may be adapted in order to maximize the science reach of big-data projects. The most successful approach to date consists of combining deep learning with innovative and extreme scale computing.

Deep learning was first proposed as a novel signal-processing tool for gravitational wave astrophysics in [14]. That initial approach considered a 2D signal manifold for binary black hole mergers, namely the masses of the binary components ( $m_1, m_2$ ), and considered simulated advanced LIGO noise. The fact that such method was as sensitive as template-matching algorithms, but at a fraction of the computational cost and orders of magnitude faster, provided sufficient motivation to extend such methodology and apply it to detect real gravitational wave sources in advanced LIGO noise in [15, 16]. These studies have sparked the interest of the gravitational wave community to explore the use of deep learning for the detection of the large zoo of gravitational wave sources [17–38].

Deep learning methods have matured to now cover a 4D signal manifold that describes the masses of the binary components and the  $z$ -component of the 3D spin vector: ( $m_1, m_2, s_1^z, s_2^z$ ) [39, 40]. These algorithms have been used to search for and find gravitational wave sources processing open source advanced LIGO data in bulk, which is available at the Gravitational Wave Open Science Center [41]. In the context of multi-messenger sources, deep learning has been used to forecast the merger of binary neutron stars and black hole-neutron star systems [37, 42]. The importance of including eccentricity for deep learning forecasting has also been studied and quantified [38]. In brief, deep learning research is moving at an incredible pace.

Another application area that has gained traction is the use of deep learning for gravitational wave parameter estimation. The established approach to estimate the astrophysical parameters of gravitational wave signals is through Bayesian inference [43–46], which is a well tested and extensively used method, though computationally-intensive. On the other hand, given the scalability and computational efficiency of deep learning models, the gravitational wave parameter estimation can take advantage of its power to produce faster inference.

Gravitational wave parameter estimation has rapidly evolved from point-wise parameter estimation [14–16] to the use of neural networks dropouts to provide estimation intervals [47], and to output a parametrized approximation of the corresponding posterior distribution [48]. Other methods have proposed the use of Conditional Variational Auto-Encoders (CVAEs) to infer the parameters of GWs embedded in simulated noise [49, 50]. In [51] the authors harnesses new methods, e.g. normalizing flow [52], to do parameter estimation over the full 15-dimensional space of binary black hole system parameters for the event GW150914. Building upon this study, authors in [53] presented deep learning methods to estimate the astrophysical parameters of several gravitational wave events. One can also refer to [54, 55] for a comprehensive review of the gravitational-wave-based machine learning approaches.

In this article we quantify the ability of deep learning to estimate the masses of the binary components of binary black hole mergers, and of the astrophysical parameters that describe the properties of the black hole remnant, namely, the final spin,  $a_f$ , and the frequency and damping time of the ringdown oscillations of the fundamental  $\ell = m = 2$  bar mode,  $(\omega_R, \omega_I)$ , known as quasinormal modes (QNMs) [56]. An existing approach proposes to use neural networks to solve differential equations for QNMs [57]. Our approach, on the other hand, differs from this or other studies in the literature in that we estimate the astrophysical parameters of the remnant by directly feeding time-series advanced LIGO strain data into our deep learning algorithms.

This article is organized as follows. In section 2 we describe the architecture of our neural network model, and the datasets used to train, validate and test it. We briefly describe the Bayesian inference pipeline, PyCBC Inference, in section 3, which we used as a baseline to compare the full posterior distributions predicted by our deep learning model. We quantify the accuracy and physical consistency of the predictions of our deep learning model for several gravitational wave sources in section 4. We summarize our findings and future directions of work in section 5.

## 2. Methods

Herein we describe several methods to improve the training performance and model accuracy of our algorithms. We have used PyTorch [58] to design, train, validate and test our neural network models.

### 2.1. Deep learning model objective

The goal of our deep learning model is to estimate a posterior distribution of the physical parameters of the waveforms from the input noisy data. This approach shares similarities with Bayesian approaches such as

Markov Chain Monte Carlo (MCMC), e.g. once a likelihood function and a predefined prior are provided, posterior samples may be drawn. The difference between the deep learning model and MCMC is that our proposed framework will learn a distribution from which we can easily draw samples, thereby increasing computational efficiency significantly. It is worth emphasizing that once the likelihood model is properly defined, the framework we introduce here may be applicable to other disciplines.

In the context of gravitational waves, the noisy waveform  $y$  is generated according to the following physical model,

$$y_{i,\ell} = F(x_i) + n_{i,\ell}, \quad (1)$$

where  $F$  is the function that maps the physical parameters (masses and spins)  $x_i$  to the gravitational waveform template [46, 59, 60], and  $n_{i,\ell}$  denotes the additive noise at various signal-to-noise ratios (SNR). We use  $y_{i,\ell}$  with subscript pair  $(i, \ell)$  to specifically indicate the  $i$ th template associated with  $\ell$ th noise realization in our dataset  $D$ . For simplicity, we use  $y$  and  $x$  to indicate noisy waveforms and the physical parameters when the specification of  $i$  or  $\ell$  subscript is not needed. We use  $K$  and  $M$  to denote the dimension of  $y$  and  $x$ , respectively.

We use WaveNet [61] to extract features from the input noisy waveforms. WaveNet was first introduced as an audio synthesis tool to generate human-like audios given random inputs. It uses dilated convolutional kernel and residual network to capture the spatial information both in the time domain and the model depth, which has been shown to be a powerful tool in model time-series data. Previously, [40, 62] tailored this architecture for gravitational wave denoising and detection. The encoded feature vector  $h \in \mathbb{R}^L$  comes from an embedding function parameterized by the WaveNet weights  $\omega$ ,  $f_\omega : y \mapsto h$ . In other words,  $h = f_\omega(y)$ .

Normalizing flow is a technique to transform distributions with invertible parameterized functions. Specifically, we use a conditional version of normalizing flow: conditional autoregressive spline [63–66] to learn the posterior distribution on top of the encoded latent space by WaveNet encoding, and we implement it through a PyTorch-based probabilistic programming package: Pyro [65]. Mathematically, we denote the invertible function  $g_{(h,\theta)} : z \mapsto x$  is parameterized by the learnable model weights  $\theta$  and the encoded feature  $h$ . In this way, we encode dependencies of the posterior distribution on the input  $y$ . The random vector  $z \in \mathbb{R}^M$  is drawn according to a pre-defined base distribution  $p(z)$ , and has the same dimension as  $x$ . The function  $g_{(h,\theta)}(z)$  is then used to convert the base distribution  $p(z)$  to the approximated posterior distribution  $\hat{p}_{\omega,\theta}(x|y)$  of the physical parameters,

$$\hat{p}_{\omega,\theta}(x|y) = p(z) \left| \det \left( \frac{\partial g_{(h,\theta)}(z)}{\partial z} \right) \right|^{-1}, \quad (2)$$

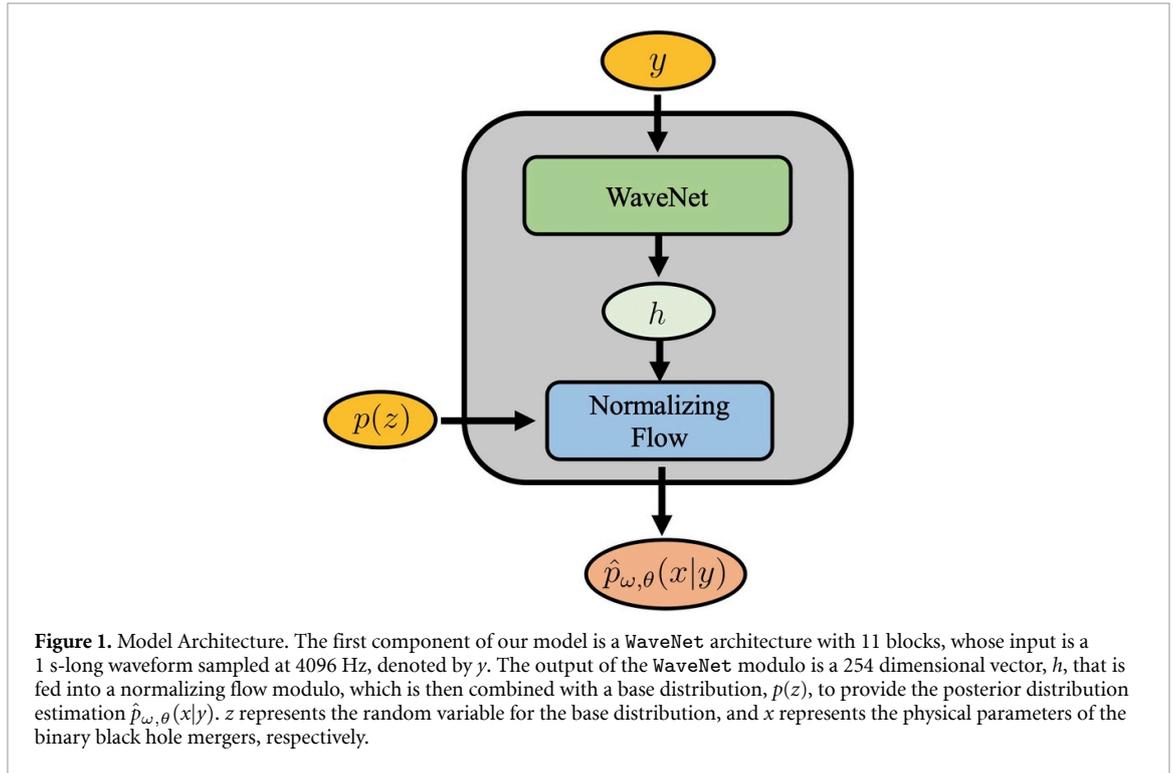
with  $h = f_\omega(y)$ .

The computation of the transformation  $g_{(h,\theta)}(z)$  contains two steps. The first step is to compute the intermediate coefficients  $\alpha$  from the feature vector  $h$  based on the function  $k_\theta$ , which is parameterized by two fully connected layers with weights denoted as  $\theta$ , i.e.  $\alpha = k_\theta(h)$ . The coefficients  $\alpha$  are used to combine the invertible linear rational splines to form  $g_{(h,\theta)}$  (see equation (5) in [64] for details). Therefore,  $g_{(h,\theta)}$  is an element-wise invertible linear rational spline with coefficients  $\alpha$ . Since  $h$  depends on the input waveform  $y$  and  $\alpha = k_\theta(h)$ , the resulting mapping  $g_{(h,\theta)}$  and parameterized distribution in equation (2) vary with the input  $y$ . The parameterization of the estimated posterior distribution is illustrated in figure 1.

To learn the network weights, we need to construct the empirical loss objective given the collection of training data  $\{x_i, y_{i,\ell}\}$ . We propose to include a loss term defined on the feature vectors in our learning objective to take account for the variation in the waveform due to noise. That is if the underlying physical parameters are similar, then the similarity of the feature vectors should be large, and vice versa. To achieve this, we use contrastive learning objective [67] to distinguish positive data pairs (waveforms with the same physical parameters) from the negative pairs (noisy waveforms with different physical parameters). Specifically, we use the normalized temperature-scaled cross entropy (NT-Xent) loss used in the state-of-the-art contrastive learning technique SimCLR [68, 69]. SimCLR was originally introduced to improve the performance of image classification with additional data augmentation and NT-Xent loss evaluation. We adapt the NT-Xent loss used in contrastive learning to our feature vectors,

$$l(h_{i,j}, h_{i,\ell}) \equiv -\log \frac{e^{\text{sim}(h_{i,j}, h_{i,\ell})/\tau}}{\sum_{i' \neq i} \sum_{j=1}^2 \sum_{\ell=1}^2 e^{\text{sim}(h_{i,j}, h_{i',\ell})/\tau}}, \quad (3)$$

where  $h_{i,\cdot} = f_\omega(y_{i,\cdot})$ ,  $\tau \in (0, \infty)$  is a scalar temperature parameter, and we choose  $\tau = 0.2$  according to the default setting provided in [68]. The NT-Xent loss performs in such a way that, regardless of the noise statistics, the cosine distances of the encoded features associated with the same underlying physical



parameters (i.e.  $h_{i,j}$  and  $h_{i,\ell}$ ) are minimized, and the distances of features with different underlying physical parameters are maximized. Consequently, the trained model is robust to the change of noise realizations and noise statistics. Therefore, incorporating the term in equation (3) can be used as a noise stabilizer for gravitational wave parameter estimation. We found that the inclusion of this term speeds up the convergence in training.

Our deep learning objective in equation (4) combines the NT-Xent loss in equation (3) with the posterior approximation term. Given a batch of  $B$  physical parameters  $x_i$ , we generate different noise realizations  $y_{i,\ell}$  for each  $x_i$  and the empirical loss function is,

$$L(\omega, \theta) = \frac{1}{2B} \sum_{i=1}^B \left[ - \sum_{\ell=1}^2 \log \hat{p}_{\omega, \theta}(x_i | y_{i,\ell}) + \sum_{\ell=1, \ell \neq j}^2 \sum_{j=1}^2 I(f_{\omega}(y_{i,j}), f_{\omega}(y_{i,\ell})) \right], \quad (4)$$

where  $\hat{p}_{\omega, \theta}(x_i | y_{i,\ell})$  is defined in equation (2). Minimizing the loss in equation (4) with respect to  $\omega$  and  $\theta$  provides a posterior estimation for gravitational wave events.

It is worth pointing out that while references [70, 71] apply  $q(z)$ , an arbitrary random distribution to their generative model, our posterior distributions do not involve arbitrary random distributions.

## 2.2. Separate models for parameters

In this paper, we are interested in the following physical parameters:  $(m_1, m_2, a_f, \omega_R, \omega_I)$ . We find that trying to estimate all parameters using a single model lead to sub-optimal results given that they are of different scales. Thus, we use two separate models with similar model architecture as shown in figure 1. One model is used to estimate the masses  $(m_1, m_2)$  of the binary components, while the other one is used to infer the final spin  $(a_f)$  and QNMs  $(\omega_R, \omega_I)$  of the remnant.

The final spin of the remnant and its QNMs have similar range of values when the QNMs are cast in dimensionless units. We trained the second model using the fact that the QNMs are determined by the final spin  $a_f$  using the relation [56]:

$$\omega_{220}(a_f) = \omega_R + i\omega_I, \quad (5)$$

where  $(\omega_R, \omega_I)$  correspond to the frequency and damping time of the ringdown oscillations for the fundamental  $\ell = m = 2$  bar mode, and the first overtone  $n = 0$ . We compute the QNMs following [56]. One can translate  $\omega_R$  into the ringdown frequency (in units of Hertz) and  $\omega_I$  into the corresponding (inverse)

damping time (in units of seconds) by computing  $M_f \cdot \omega_{220}$ , where  $M_f$  is the final mass of the remnant, and can be determined using equation (1) in [72]. An additional benefit of using two separate models is that the training converges faster with two models considering two different sets of physical parameters at different magnitudes.

## 2.3. Dataset preparation and training

### 2.3.1. Modeled waveforms

We used the surrogate waveform family [73] to produce modeled waveforms that describe binary black holes with component masses  $m_1 \in [10M_\odot, 80M_\odot]$ ,  $m_2 \in [10M_\odot, 50M_\odot]$ , and spin components  $s_{\{1,2\}}^z \in [-0.9, 0.9]$ . By uniformly sampling this parameter space we produce a dataset with 1061 023 waveforms. These waveforms describe the last second of the late inspiral, merger, and ringdown. The waveforms are produced using a sample rate of 4096 Hz.

For training purposes, we label the waveforms using the masses and spins of the binary components, and then use this information to also enable the neural net to estimate the final spin of the black hole remnant using the formulae provided in [74], and the QNMs following [56]. In essence, we are training our neural network models to identify the key features that determine the properties of the binary black holes before and after merger using a unified framework.

We use 90% of these waveform samples for training, 10% testing. The training samples are randomly and uniformly chosen. Throughout the training, we use AdamW optimizer to minimize the mean squared error of the predicted parameters with default hyper-parameter setups [75]. We choose the learning rate to be 0.0001. To simulate the environment where the true gravitational waves are embedded, we use real advanced LIGO noise to compute power spectral density (PSD), which is then used to whiten the templates.

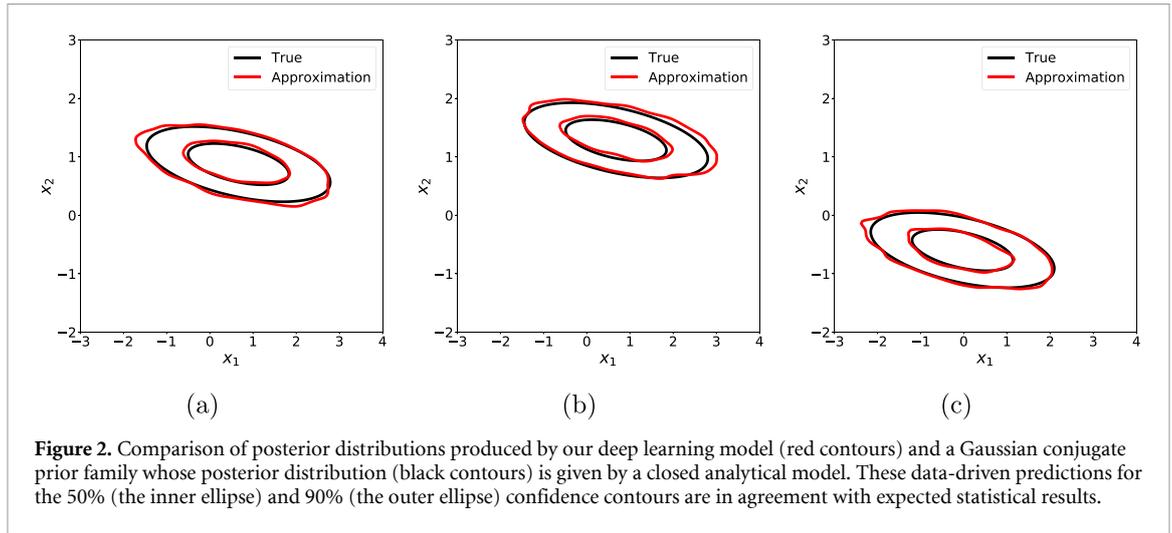
### 2.3.2. Advanced LIGO noise

For training we used a 4096 s-long advanced LIGO noise data segment, sampled at 4096 Hz, starting at GPS time 11 262 59 462. We obtained these data from the Gravitational Wave Open Science Center [41]. We estimate a PSD using the entire 4096 s segment to whiten the modeled waveforms and noise. For each one second long noisy waveform used in training, we combine the clean whitened template with a randomly picked one second long noise segment from the 4096 s-long advanced LIGO strain data. For each generated waveform template (see equation (1)), we apply two different noisy realizations. As a result, the total number of noisy waveforms (clean templates + noise realizations) applied during training is equal to: # of training iterations  $\times$  batch size  $\times$  2.

In section 4, we demonstrate that our model, trained only with advanced LIGO noise from the first observing run, is able to estimate the astrophysical parameters of other events across O1-O3. We fixed the merger point of the training templates at the 3596th timestep out of 4096 total timesteps. We empirically found having a fixed merger point, rather than shifting the templates to have time-invariant property, provides a tighter estimation of the posteriors. Our deep learning model was trained on 1 NVIDIA V100 GPU with a batch size of 8. In general, it takes about one to two days to fully train this model.

## 2.4. GPS trigger time

It is known that a trigger GPS time associated with a gravitational wave event, typically provided by a detection algorithm, may differ from the true time of coalescence. Therefore, we perform a local search around the trigger time by any given detection algorithm as a pre-processing step for the parameter estimation using the trained model. We first identify local merger time candidates by evaluating the normalized cross-correlation (NCC) of the whitened observation with 33 713 whitened clean templates, whose physical parameters uniformly cover the range:  $m_1 \in [10M_\odot, 80M_\odot]$ ,  $m_2 \in [10M_\odot, 50M_\odot]$ , and  $s_{\{1,2\}}^z \in [-0.9, 0.9]$ , over a time window of 0.015 seconds around the time candidates. The time points with top NCC values are selected as the candidates. Then we use the trained models to estimate the posterior distributions of the physical parameters at each candidate time point. In practice, we found that the trigger times with the best NCC values differ from those published at the Gravitational Wave Open Science Center by up to 0.01 s. These trigger times produce different posterior distributions that vary in size by up to  $\pm 1M_\odot$  for the masses of the binary components, and up to 5% for the astrophysical properties of the compact remnant. We have selected the time point that gives the smallest 90% confidence area for the results we present in section 4.2.



**Figure 2.** Comparison of posterior distributions produced by our deep learning model (red contours) and a Gaussian conjugate prior family whose posterior distribution (black contours) is given by a closed analytical model. These data-driven predictions for the 50% (the inner ellipse) and 90% (the outer ellipse) confidence contours are in agreement with expected statistical results.

### 3. Bayesian inference

We compare our data-driven posterior estimation with PyCBC Inference [46, 59, 60], which uses a parallel-tempered MCMC algorithm, emcee\_pt [76], to evaluate the posterior probability  $p(x|y)$  for the set of source parameters  $x$  given the data  $y$ . The posterior is calculated as  $p(x|y) \propto p(y|x)p(x)$  where  $p(y|x)$  is the likelihood and  $p(x)$  is the prior. The likelihood function for a set of  $N$  detectors is

$$p(y|x) = \exp\left(-\frac{1}{2} \sum_{i=1}^N \langle \hat{y}_i(k) - \hat{s}_i(k, x) | \hat{y}_i(k) - \hat{s}_i(k, x) \rangle\right), \tag{6}$$

where  $\hat{y}_i(k)$  and  $\hat{s}_i(k, x)$  are the frequency-domain representations of the data and the model waveform for detector  $i$ . The inner product  $\langle \cdot | \cdot \rangle$  is defined as

$$\langle \hat{a}_i(k) | \hat{b}_i(k) \rangle = 4 \mathcal{R} \int_0^\infty \frac{\hat{a}_i(k) \hat{b}_i^*(k)}{P^i(k)} dk, \tag{7}$$

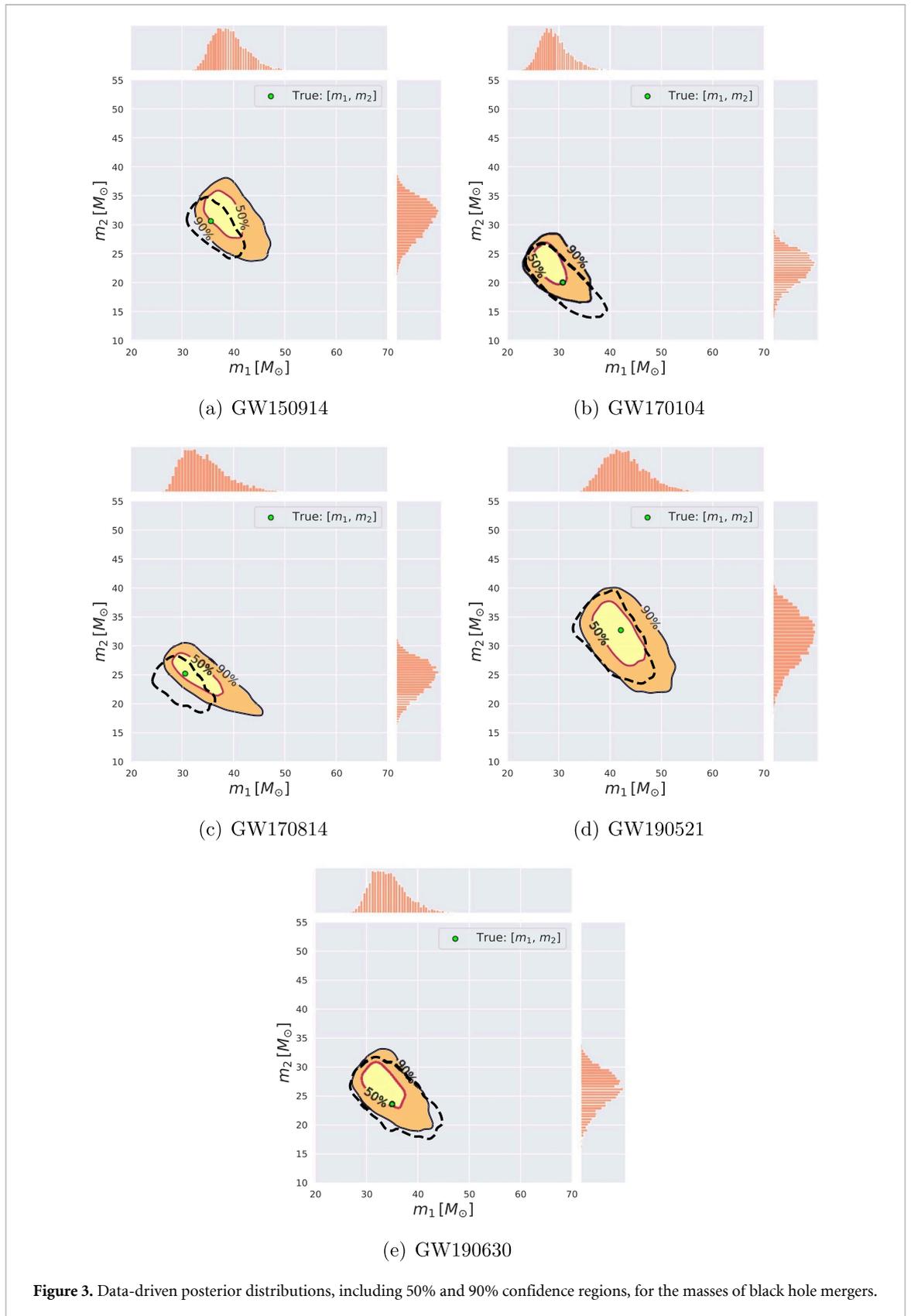
where  $P^i(k)$  is the PSD of the  $i$ th detector.

We performed the MCMC analysis using the publicly available data from the GWTC-1 release [4] and used the corresponding publicly available PSD files for each event [77]. We analyse a segment of eight seconds around the GPS trigger 11 675 59 935.6, with the data sampled to 2048 Hz. We use the IMRPhenomD [78] waveform model to generate waveform templates to evaluate the likelihood. We assume uniform priors for the component masses with  $m_{\{1,2\}} \in [10M_\odot, 80M_\odot]$  and uniform priors on the component spins with  $a_{\{1,2\}} \in (-0.99, 0.99)$ . We also set uniform priors on the luminosity distance with  $D_L \in [10, 4000]\text{Mpc}$  and the deviation of the arrival time from the trigger time  $-0.1 < \Delta t < 0.1$ . We set uniform priors for the coalescence phase and the polarization angle  $\phi_c, \psi \in [0, 2\pi)$ . The prior on the inclination angle between the binary’s orbital angular momentum and the line of sight,  $\iota$ , is set to be uniform in the sine of the angle, and the right ascension and declination have priors to be uniform over the sky.

Furthermore, they may be used to cross validate the physical reality of an event [39, 40], and to assess whether the estimated merger time is consistent between the two separate models. For instance, if the models output very different merger times, then we may conclude that they are not providing a reliable merger time. On the other hand, when their results are consistent, within a window between 0.001 s and up to 0.005 s, then we can remove the ambiguity introduced when using the NCC approach described in section 2.4.

### 4. Experimental results

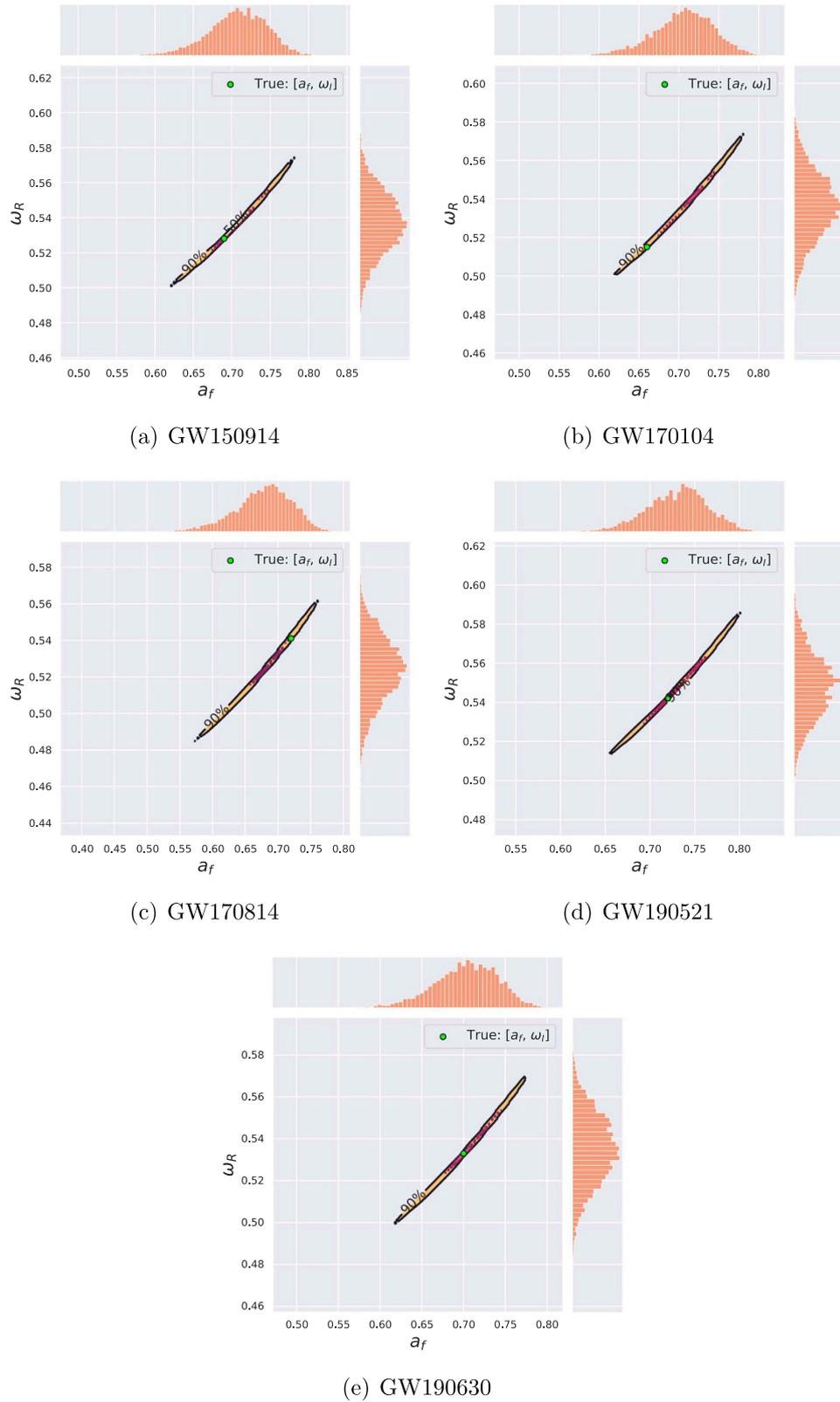
In this section we present two types of results. First, we validate our model with a well known statistical model. Upon confirming that our deep learning approach is statistically consistent, we used to estimate the parameters of five binary black hole mergers.



#### 4.1. Validation on simulated data

We performed experiments on simulated data that have closed form posterior distributions. This is important to ascertain the accuracy and reliability of our method. The simulated data are generated through a linear observation model with additive white Gaussian noise,

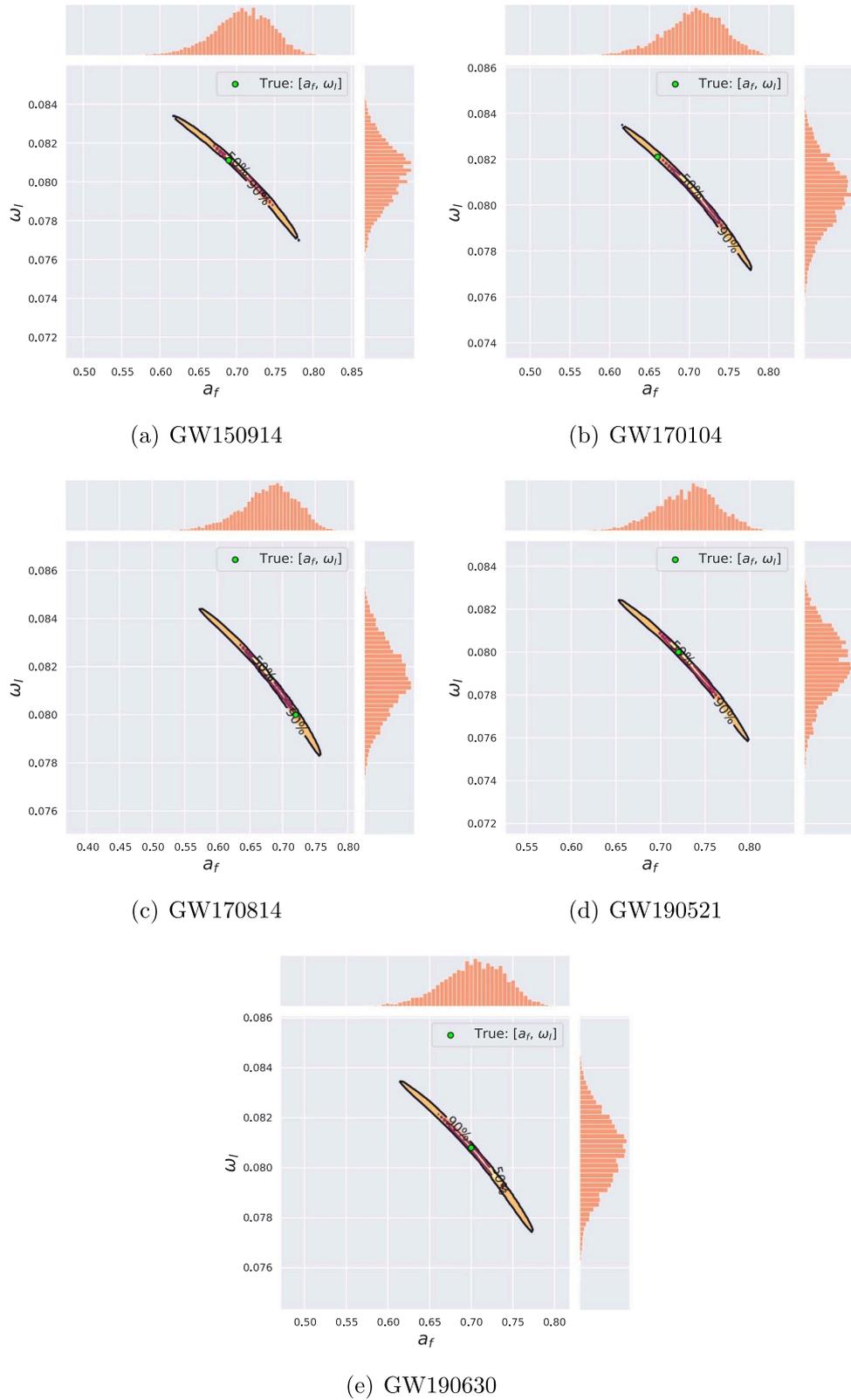
$$y = Ax + n, \tag{8}$$



**Figure 4.** Data-driven posterior distributions, including 50% and 90% confidence regions, for  $(a_f, \omega_R)$  of black hole mergers.

where the additive noise  $n \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)$ . We consider the underlying parameters  $x \in \mathbb{R}^M$  and the linear map  $A \in \mathbb{R}^{K \times M}$ , with  $M = 2$  and  $K = 5$ . The likelihood function is

$$p(y|x) = \frac{1}{(\sqrt{2\pi}\sigma)^K} \exp\left(-\frac{\|y - Ax\|^2}{2\sigma^2}\right). \tag{9}$$



**Figure 5.** Data-driven posterior distributions, including 50% and 90% confidence regions, for  $(a_f, \omega_l)$  of real black hole mergers.

If we assume the prior distribution of  $x$  is a Gaussian distribution with mean  $\mathbf{0}$  and covariance  $S$ , we can get an analytical expression for the posterior distribution of  $x$  given the observation  $y$ ,

$$p(x|y) = C \exp\left(-\frac{1}{2}(y - \Sigma^{-1}b)^T \Sigma^{-1}(y - \Sigma^{-1}b)\right), \tag{10}$$

**Table 1.** Data-driven and Bayesian results [4, 79] for the median and 90% confidence intervals of the masses of five binary black hole mergers. The network signal-to-noise ratio (SNRs) for each event are also provided for reference.

Event name	Our model		LIGO [4, 79]		SNR
	$m_1$ [ $M_\odot$ ]	$m_2$ [ $M_\odot$ ]	$m_1$ [ $M_\odot$ ]	$m_2$ [ $M_\odot$ ]	
GW150914	$38.85^{+6.90}_{-4.15}$	$31.20^{+4.39}_{-5.94}$	$35.60^{+4.70}_{-3.10}$	$30.60^{+3.00}_{-4.40}$	24.4
GW170104	$28.90^{+6.55}_{-3.80}$	$22.75^{+3.73}_{-5.14}$	$30.80^{+7.80}_{-5.60}$	$20.00^{+4.90}_{-4.60}$	13.0
GW170814	$33.92^{+9.14}_{-5.27}$	$24.31^{+4.13}_{-5.46}$	$30.60^{+5.60}_{-5.30}$	$25.20^{+2.80}_{-4.00}$	15.9
GW190521	$46.10^{+8.77}_{-6.61}$	$33.74^{+6.68}_{-8.47}$	$42.10^{+5.90}_{-4.90}$	$32.70^{+5.40}_{-6.20}$	14.4
GW190630	$34.00^{+7.19}_{-4.43}$	$26.17^{+4.54}_{-5.86}$	$35.00^{+6.90}_{-5.70}$	$23.60^{+5.20}_{-5.10}$	15.6

where

$$C = \frac{1}{\sqrt{(2\pi)^M |\Sigma|}}, \quad \Sigma = S^{-1} + \frac{1}{\sigma^2} A^T A, \quad b = \frac{1}{\sigma^2} A^T y.$$

During the training stage we draw 100 samples of  $x$  from its prior  $p(x)$ , and  $y$  is generated through the linear observation model (8). We train a 3-layer model with the model objective (4), and show three examples of the posterior estimation in figure 2. Therein we show 50% and 90% confidence contours. Black lines represent ground truth results (ellipses given the posterior is Gaussian), while the red contours correspond to the neural network estimations, based on Gaussian kernel density estimation (KDE) with 9000 samples generated from the network. These results indicate that our deep learning model can produce reliable and statistically valid results.

#### 4.2. Results with real events

In this section we use our deep learning models to estimate the medians and posterior distributions of the astrophysical parameters  $(m_1, m_2)$  and  $(a_f, \omega_R, \omega_I)$ , respectively, for five binary black hole mergers: GW150914, GW170104, GW170814, GW190521 and GW190630.

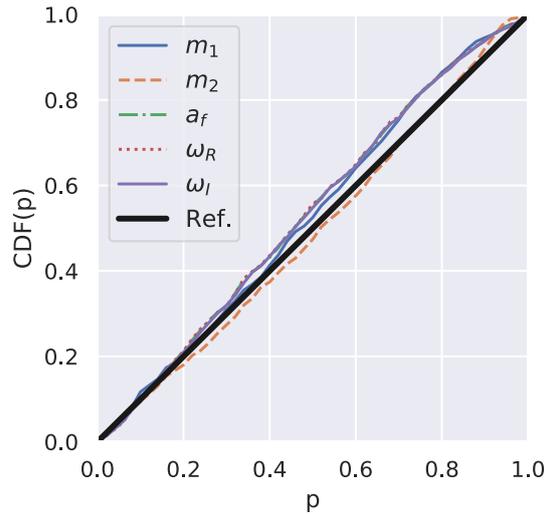
As described in section 2.1, we consider 1 s-long advanced LIGO noise input data batches, denoted as  $y$ , sampled at 4096 Hz. We construct two posterior distribution estimations,  $\hat{p}_{\omega, \theta}(x|y)$ , by minimizing the loss in equation (4) for  $(m_1, m_2)$  and for  $(a_f, \omega_R, \omega_I)$ . We use two different multivariate normal base distributions for  $p(z)$  in the two different models. To estimate the masses of the binary components, the mean and covariance matrix  $(\mu, \Sigma)$  are:  $\mu = (30, 30)$ ,  $\Sigma = \text{diag}(5, 5)$ ; whereas for the final spin and QNMs model we use:  $\mu = (0.5, 0.55, 0.07)$ ,  $\Sigma = \text{diag}(0.05, 0.03, 0.002)$ . ‘diag( $\cdot$ )’ refers to the diagonal matrix with ‘ $\cdot$ ’ being the diagonal elements. The number of normalizing flow layers also varies for the two models. We use a 3-layer normalizing flow module for masses prediction, and an 8-layer module for the predictions of final spin and QNMs.

Our first set of results is presented in figures 3–5. These figures provide the median, and the 50% and 90% confidence intervals, which we computed using Gaussian KDE estimation with 9000 samples drawn from the estimated posteriors. In tables 1 and 2 we also present a summary of our data-driven median results and 90% confidence intervals, along with those obtained with traditional Bayesian algorithms in [4, 79]. Before we present the main highlights of these results, it is important to emphasize that our results are entirely data-driven. We have not attempted to use deep learning as a fast interpolator that learns the properties of traditional Bayesian posterior distributions. Rather, we have allowed deep learning to figure out the physical correlations among different parameters that describe the physics of black hole mergers. Furthermore, we have quantified the statistical consistency of our approach by validating it against a well known model. This is of paramount importance, since deep learning models may be constructed to reproduce the properties of traditional Bayesian distributions, but that fact does not provide enough evidence of their statistical validity or consistency. Finally, given the nature of the signal processing tools and computing approaches we use in this study, we do not expect our data-driven results to exactly reproduce the traditional Bayesian results reported in [4, 79].

Our results may be summarized as follows. Figures 3–5 show that our data-driven posterior distributions encode expected physical correlations for the masses of the binary components,  $(m_1, m_2)$ , and the parameters of the remnant:  $(a_f, \omega_R)$  and  $(a_f, \omega_I)$ . We also learn that these posterior distributions are determined by the properties of the noise and loudness of the signal that describes these events. Figure 3 presents a direct comparison between the posterior distributions predicted by our deep learning models and those produced with PyCBC Inference—marked with dashed lines. These results show that our deep learning models provide real-time, reliable information about the astrophysical properties of binary black hole mergers that were detected in three different observing runs, and which span a broad SNR range.

**Table 2.** Data-driven and Bayesian results [4, 79] for the median and 90% confidence intervals of the final spin of five binary black hole mergers. Results for the frequencies of the ringdown oscillations,  $(\omega_I, \omega_R)$ , are directly measure by our model from advanced LIGO's strain data, whereas the results quoted for LIGO are estimated using  $a_f$  values from [4, 79] and equation (5) [80].

Event name	Our model			LIGO [4, 79]		
	$a_f$	$\omega_R$	$\omega_I$	$a_f$	$\omega_R$	$\omega_I$
GW150914	$0.71^{+0.06}_{-0.07}$	$0.536^{+0.028}_{-0.029}$	$0.0805^{+0.0023}_{-0.0026}$	$0.69^{+0.05}_{-0.04}$	$0.528^{+0.016}_{-0.023}$	$0.0811^{+0.0021}_{-0.0013}$
GW170104	$0.69^{+0.06}_{-0.07}$	$0.530^{+0.028}_{-0.030}$	$0.0810^{+0.0023}_{-0.0025}$	$0.66^{+0.08}_{-0.11}$	$0.515^{+0.036}_{-0.033}$	$0.0821^{+0.0026}_{-0.0030}$
GW170814	$0.68^{+0.06}_{-0.09}$	$0.525^{+0.028}_{-0.032}$	$0.0815^{+0.0024}_{-0.0024}$	$0.72^{+0.07}_{-0.05}$	$0.541^{+0.037}_{-0.022}$	$0.0800^{+0.0018}_{-0.0037}$
GW190521	$0.73^{+0.05}_{-0.06}$	$0.548^{+0.029}_{-0.028}$	$0.0795^{+0.0024}_{-0.0029}$	$0.72^{+0.05}_{-0.07}$	$0.552^{+0.026}_{-0.030}$	$0.0800^{+0.0025}_{-0.0025}$
GW190630	$0.71^{+0.06}_{-0.07}$	$0.535^{+0.028}_{-0.030}$	$0.0806^{+0.0024}_{-0.0026}$	$0.70^{+0.06}_{-0.07}$	$0.532^{+0.030}_{-0.028}$	$0.0808^{+0.0037}_{-0.0022}$



**Figure 6.** P-P plot comparing the posterior distributions estimated by the neural network model for five astrophysical parameters  $(m_1, m_2, a_f, \omega_R, \omega_I)$ .

On the other hand, tables 1 and 2 show that our median and 90% confidence intervals are better, similar and in some cases slightly larger than those obtained with Bayesian algorithms. In these tables, Bayesian LIGO results for  $a_f$  are directly taken from [4, 79], while  $(\omega_R, \omega_I)$  results are computed using their Bayesian results for  $a_f$  and the tables available at [81]. These results indicate that deep learning methods can *learn physical correlations in the data*, and provide reliable estimates of the parameters of gravitational wave sources. To demonstrate that our model represents true statistical properties of the posterior distribution, we tested the posterior estimation on simulated noisy gravitational waveforms. We calculate the empirical cumulative distribution function (CDF) of the number of times the true value for each parameter was found within a given confidence interval  $p$ , as a function of  $p$ . We compare the empirical CDF with the true CDF of  $p$  in the P-P plot in figure 6. To obtain the empirical CDF, for each test waveform (1000 waveforms in total) and one-dimensional estimated posterior distribution generated from the network with 9000 samples, we record the count of the confidence intervals  $p$  ( $p = 1\% \dots, 100\%$ ) where the true parameters fall. The empirical CDF is based on the frequency of such counts with the 1000 waveforms randomly drawn from the test dataset. Since the empirical CDFs lie close to the diagonal, we conclude that the networks generate close approximation of the posteriors. Furthermore, our data-driven results, including medians and posterior distributions, can be produced *within 2 milliseconds per event* using a single NVIDIA V100 GPU. We expect that these tools will provide the means to assess in real-time whether the inferred astrophysical parameters of the binary components and the post-merger remnant adhere to general relativistic predictions. If not, these results may prompt follow up analyses to investigate whether apparent discrepancies are due to poor data quality or other astrophysical effects [82].

The reliable astrophysical information inferred in low-latency by deep learning algorithm warrants the extension of this framework to characterize other sources, including eccentric compact binary mergers, and sources that require the inclusion of higher-order waveform modes. Furthermore, the use of physics-inspired deep learning architectures and optimization schemes [29] may enable an accurate measurement of the spin of binary components. These studies should be pursued in the future.

## 5. Conclusion

We designed neural networks to estimate five parameters that describe the astrophysical properties of binary black holes before and after the merger event. The first two parameters constrain the masses of the binary components, while the others estimate the properties of the black hole remnant, namely  $(m_1, m_2, a_f, \omega_R, \omega_1)$ . These models combine a WaveNet architecture with normalizing flow and contrastive learning to provide statistically consistent estimates for both simulated distributions, and real gravitational wave sources.

Our findings indicate that deep learning can abstract physical correlations in complex data, and then provide reliable predictions for the median and 90% confidence intervals for binary black holes that span a broad SNR range. Furthermore, while these models were trained using only advanced LIGO noise from the first observing run, they were capable of generalizing to binary black holes that were reported during the first, second and third observing runs.

These models will be extended in future work to provide informative estimates for the spin of the binary components, including higher-order waveform modes to better model the physics of highly spinning and asymmetric mass-ratio black hole systems.

## Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: <https://www.gw-openscience.org/about/>.

## Acknowledgments

Neural network models are available at the Data and Deep Learning Hub for Science [83, 84]. E A H, H S and Z Z gratefully acknowledge National Science Foundation (NSF) awards OAC-1931561 and OAC-1934757. E O S and P K gratefully acknowledge NSF grants PHY-1912081 and OAC-193128, and the Sherman Fairchild Foundation. P K also acknowledges the support of the Department of Atomic Energy, Government of India, under Project No. RTI4001. This work utilized the Hardware-Accelerated Learning (HAL) cluster, supported by NSF Major Research Instrumentation program, Grant OAC-1725729, as well as the University of Illinois at Urbana-Champaign. Compute resources were provided by XSEDE using allocation TG-PHY160053. This work made use of the Illinois Campus Cluster, a computing resource that is operated by the Illinois Campus Cluster Program (ICCP) in conjunction with the National Center for Supercomputing Applications and which is supported by funds from the University of Illinois at Urbana-Champaign. This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357. This research also made use of LIGO Data Grid clusters at the California Institute of Technology. This research used data, software and/or web tools obtained from the LIGO Open Science Center (<https://gw-openscience.org>), a service of LIGO Laboratory, the LIGO Scientific Collaboration and the Virgo Collaboration. LIGO is funded by the U.S. National Science Foundation. Virgo is funded by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale della Fisica Nucleare (INFN) and the Dutch Nikhef, with contributions by Polish and Hungarian institutes.

## ORCID iDs

Hongyu Shen  <https://orcid.org/0000-0001-7620-805X>

E A Huerta  <https://orcid.org/0000-0002-9682-3604>

Zhizhen Zhao  <https://orcid.org/0000-0003-3594-5840>

## References

- [1] Abbott B P *et al* 2016 *Phys. Rev. Lett.* **116** 131103
- [2] Aasi J *et al* 2015 *Class. Quantum Grav.* **32** 074001
- [3] Acernese F *et al* 2015 *Class. Quantum Grav.* **32** 024001
- [4] Abbott B P *et al* 2019 *Phys. Rev. X* **9** 031040
- [5] Abbott R *et al* 2021 *Phys. Rev. X* **11** 021053
- [6] Abbott R *et al* 2021 *Astrophys. J. Lett.* **913** L7
- [7] Huerta E *et al* 2017 Boss-LDG: a novel computational framework that brings together blue waters, open science grid, shifter and the LIGO data grid to accelerate gravitational wave discovery 2017 *IEEE 13th Int. Conf. on e-Science (e-Science)* (IEEE) pp 335–44
- [8] Huerta E, Haas R, Jha S, Neubauer M and Katz D S 2019 *Comput. Softw. Big Sci.* **3** 5
- [9] Weitzel D, Bockelman B, Brown D A, Couvares P, Würthwein F and Hernandez E F 2017 Data access for LIGO on the OSG *Proc. Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact* pp 1–6

- [10] Liang S, Liu Y, Wang C and Jian L 2010 Design and evaluation of a parallel k-nearest neighbor algorithm on CUDA-enabled GPU *2010 IEEE 2nd Symp. on Web Society* pp 53–60
- [11] Allen G et al 2018 arXiv:1807.04780
- [12] Allen G et al 2019 arXiv:1902.00522
- [13] Huerta E A et al 2019 *Nat. Rev. Phys.* **1** 600–8
- [14] George D and Huerta E A 2018 *Phys. Rev. D* **97** 044039
- [15] George D and Huerta E A 2017 Deep learning for real-time gravitational wave detection and parameter estimation with LIGO data *NiPS Summer School 2017 (Gubbio, Perugia, Italy, 30 June–3 July 2017)*
- [16] George D and Huerta E 2018 *Phys. Lett. B* **778** 64–70
- [17] Gabbard H, Williams M, Hayes F and Messenger C 2018 *Phys. Rev. Lett.* **120** 141103
- [18] Skliris V, Norman M R and Sutton P J 2020 arXiv:2009.14611
- [19] Lin Y C and Wu J H P 2020 arXiv:2007.04176
- [20] Wang H, Wu S, Cao Z, Liu X and Zhu J Y 2020 *Phys. Rev. D* **101** 104003
- [21] Nakano H et al 2019 *Phys. Rev. D* **99** 124032
- [22] Fan X, Li J, Li X, Zhong Y and Cao J 2019 *Sci. China Phys. Mech. Astron.* **62** 969512
- [23] Li X R, Babu G, Yu W L and Fan X L 2020 *Front. Phys.* **15** 54501
- [24] Deighan D S, Field S E, Capano C D and Khanna G 2020 arXiv:2010.04340
- [25] Miller A L et al 2019 *Phys. Rev. D* **100** 062005
- [26] Krastev P G 2020 *Phys. Lett. B* **803** 135330
- [27] Schäfer M B, Ohme F and Nitz A H 2020 *Phys. Rev. D* **102** 063015
- [28] Dreissigacker C and Prix R 2020 *Phys. Rev. D* **102** 022005
- [29] Khan A, Huerta E and Das A 2020 *Phys. Lett. B* **808** 0370–2693
- [30] Dreissigacker C, Sharma R, Messenger C, Zhao R and Prix R 2019 *Phys. Rev. D* **100** 044009
- [31] Wei W et al 2020 *MNRAS* **493** 3178–93
- [32] Beheshtipour B and Deep P M A 2020 *Phys. Rev. D* **101** 064009
- [33] Skliris V, Norman M R K and Sutton P J 2020 arXiv:2009.14611
- [34] Khan S and Green R 2020 arXiv:2008.12932
- [35] Chua A J K, Galley C R and Vallisneri M 2019 *Phys. Rev. Lett.* **122** 211101
- [36] Rebei A, Huerta E A, Wang S, Habib S, Haas R, Johnson D and George D 2019 *Phys. Rev. D* **100** 044025
- [37] Wei W and Huerta E A 2021 *Phys. Lett. B* **816** 136185
- [38] Wei W, Huerta E A, Yun M, Loutrel N, Shaikh M A, Kumar P, Haas R and Kindratenko V 2021 *Astrophys. J.* **919** 82
- [39] Huerta E A et al 2021 *Nat. Astron.* **5** 1062–8
- [40] Wei W, Khan A, Huerta E A, Huang X and Tian M 2021 *Phys. Lett. B* **812** 136029
- [41] Vallisneri M, Kanner J, Williams R, Weinstein A and Stephens B 2015 *J. Phys.: Conf. Ser.* **610** 012021
- [42] Yu H, Adhikari R X, Magee R, Sachdev S and Chen Y 2021 arXiv:2104.09438
- [43] Graff P, Feroz F, Hobson M P and Lasenby A 2012 *MNRAS* **421** 169–80
- [44] Veitch J et al 2015 *Phys. Rev. D* **91** 042003
- [45] Singer L P and Rapid P L R 2016 *Phys. Rev. D* **93** 024013
- [46] Biwer C M, Capano C D, De S, Cabero M, Brown D A, Nitz A H and Raymond V 2019 *Astron. Soc. Pac.* **131** 024503
- [47] Alvares J D, Font J A, Freitas F F, Freitas O G, Morais A P, da Silva Nunes S, Onofre A and Torres-Forné A 2021 *Class. Quantum Grav.* **38** 155010
- [48] Chua A J K and Vallisneri M 2020 *Phys. Rev. Lett.* **124** 041102
- [49] Gabbard H, Messenger C, Heng I S, Tonolini F and Murray-Smith R 2019 arXiv:1909.06296
- [50] Green S R, Simpson C and Gair J 2020 *Phys. Rev. D* **102** 104057
- [51] Green S R and Gair J 2021 *Mach. Learn. Sci. Tech.* **2** 03LT01
- [52] Grover A, Dhar M and Ermon S 2018 Flow-GAN: combining maximum likelihood and adversarial learning in generative models *AAAI*
- [53] Dax M, Green S R, Gair J, Macke J H, Buonanno A and Schölkopf B 2021 arXiv:2106.12594
- [54] Cuoco E et al 2020 *Mach. Learn.: Sci. Technol.* **2** 011002
- [55] Huerta E A and Zhao Z 2020 *Advances in Machine and Deep Learning for Modeling and Real-Time Detection of Multi-Messenger Sources* (Singapore: Springer Singapore) pp 1–27
- [56] Berti E, Cardoso V and Will C M 2006 *Phys. Rev. D* **73** 064030
- [57] Övgün A, Sakallı İ and Mutuk H 2021 *Int. J. Geom. Methods Mod. Phys.* **18** 2150154
- [58] Paszke A et al 2017 Automatic differentiation in pytorch *NIPS Autodiff Workshop*
- [59] Nitz A et al 2020 gwastro/pycbc: Pycbc release 1.16.4 (available at: <https://doi.org/10.5281/zenodo.3904502>)
- [60] Voudsen W D, Farr W M and Mandel I 2015 *Mon. Not. R. Astron. Soc.* **455** 1919–37
- [61] van den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A and Kavukcuoglu K 2016 arXiv:1609.03499
- [62] Wei W and Huerta E 2020 *Phys. Lett. B* **800** 135081
- [63] Durkan C, Bekasov A, Murray I and Papamakarios G 2019 Neural spline flows *Advances in Neural Information Processing Systems* pp 7511–22
- [64] Dolatabadi H M, Erfani S and Leckie C 2020 arXiv:2001.05168
- [65] Bingham E et al 2019 *J. Mach. Learn. Res.* **20** 973–8
- [66] Phan D, Pradhan N and Jankowiak M 2019 arXiv:1912.11554
- [67] Hadsell R, Chopra S and LeCun Y 2006 Dimensionality reduction by learning an invariant mapping *2006 IEEE Computer Conf. on Computer Vision and Pattern Recognition (CVPR'06) vol 2 (IEEE)* pp 1735–42
- [68] Chen T, Kornblith S, Norouzi M and Hinton G 2020 A simple framework for contrastive learning of visual representations *37th Int. Conf. on Machine Learning*
- [69] Chen T, Kornblith S, Swersky K, Norouzi M and Hinton G E 2020 *Adv. Neural Inf. Process. Syst.* **33**
- [70] Dumoulin V, Belghazi I, Poole B, Lamb A, Arjovsky M, Mastropietro O and Courville A C 2017 Adversarially Learned Inference (arXiv:1606.00704)
- [71] Donahue J, Krähenbühl P and Darrell T 2017 arXiv:1605.09782
- [72] Healy J and Lousto C O 2017 *Phys. Rev. D* **95** 024037

- [73] Blackman J, Field S E, Galley C R, Szilágyi B, Scheel M A, Tiglio M and Hemberger D A 2015 *Phys. Rev. Lett.* **115** 121102
- [74] Hofmann F, Barausse E and Rezzolla L 2016 *Astrophys. J.* **825** L19
- [75] Loshchilov I and Hutter F 2017 arXiv:1711.05101
- [76] Foreman-Mackey D, Hogg D W, Lang D and Goodman J 2013 *Astron. Soc. Pac.* **125** 306
- [77] Parameter estimation sample release for GWTC-1 2019 (available at: <https://dcc.ligo.org/LIGO-P1800370/public>)
- [78] Khan S, Husa S, Hannam M, Ohme F, Pürrer M, Forteza X J and Bohé A 2016 *Phys. Rev. D* **93** 044007
- [79] Abbott R *et al* 2020 arXiv:2010.14527
- [80] Berti E, Cardoso V and Casals M 2006 *Phys. Rev. D* **73** 024013
- [81] Berti E, Buonanno A and Will C M 2005 *Phys. Rev. D* **71** 084025
- [82] Gair J R, Vallisneri M, Larson S L and Baker J G 2013 *Living Rev. Rel.* **16** 7
- [83] Blaiszik B, Ward L, Schwarting M, Gaff J, Chard R, Pike D, Chard K and Foster I 2019 *MRS Commun.* **9**
- [84] Chard R *et al* 2019 DLHub: model and data serving for science 2019 *IEEE Int. Parallel and Distributed Symp. (IPDPS)* pp 283–92