Deterministic and Bayesian Neural Networks
for Low-latency Gravitational Wave Parameter Estimation
of Binary Black Hole Mergers

Hongyu Shen,[1,2] E. A. Huerta,[1,3] Zhizhen Zhao,[1,2,4] Elise Jennings,[5] and Himanshu Sharma[5]

[1]*NCSA, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA*
[2]*Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA*
[3]*Department of Astronomy, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA*
[4]*Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA*
[5]*Argonne Leadership Computing Facility, Argonne National Laboratory, 9700 South Cass Avenue, Lemont, Illinois 60439, USA*
(Dated: September 17, 2019)

We present the first application of deep learning for gravitational wave parameter estimation of binary black hole mergers evolving on quasi-circular orbits with aligned or anti-aligned spins. We use root-leaf structured networks to ensure that common physical features are shared across all parameters. In order to cover a broad range of astrophysically motivated scenarios, we use a training dataset with over 10^7 modeled waveforms to ensure local time- and scale-invariance. The trained models are applied to estimate the astrophysical parameters of the existing catalog of detected binary black hole mergers, and their corresponding black hole remnants, including the final spin and the gravitational wave quasi-normal frequencies. Using a deterministic neural network model, we are able to efficiently provide point-parameter estimation results, along with statistical errors caused by the noise spectrum uncertainty. We also introduce the first application of Bayesian neural networks for gravitational wave parameter estimation of real astrophysical events. These probabilistic models were trained with over 10^7 modeled waveforms and using 1024 nodes (65,536 core processors) on the Theta supercomputer at Argonne Leadership Computing Facility to reduce the training stage to just thirty minutes. In inference mode, both the deterministic and Bayesian neural networks estimate the astrophysical parameters of binary black hole mergers within 2 milliseconds using a single Tesla V100 GPU. Both deterministic and Bayesian neural networks produce agreeing parameter estimation results, which are also consistent with Bayesian analyses used to characterize the catalog of binary black hole mergers observed by the advanced LIGO and Virgo detectors.

I. INTRODUCTION

Gravitational wave (GW) sources [1–6] are now routinely detected by the advanced LIGO [7, 8] and Virgo [9] detectors. The last two observing runs of these GW detectors indicate that, on average, one GW source has been detected for every fifteen days of analyzed data. It is expected that this number will be superseded in the upcoming third observing run, since the advanced LIGO and Virgo detectors have been undergoing commissioning since August 2017. In their enhanced sensitivity configuration, they will be able to probe a larger volume of space, thereby boosting the expected detection rate for binary black hole (BBH) mergers and binary neutron star (BNS), and may yield the first observations of neutron star-black hole (NSBH) mergers [6].

Given the expected scale of GW discovery in upcoming observing runs, it is in order to explore the use of efficient signal-processing algorithms for low-latency GW detection and parameter estimation. This work is motivated by the need to probe a deeper parameter space that is available to GW detectors, in real-time, and using minimal computational resources to maximize the number of studies that can be conducted with GW data. This combination of constraints is a common theme for large-scale astronomical facilities, which will be producing large datasets in low-latency within the next decade, e.g., the Large Synoptic Survey Telescope [10]. Scenarios

in which both LSST, among other electromagnetic observatories, and advanced LIGO and Virgo work in unison, analyzing disparate datasets in real-time to realize the science goals of Multi-Messenger Astrophysics make this work timely and relevant [11, 12].

Among a number of recent developments in signal-processing, deep learning exhibits great promise to increase the speed and depth of real-time GW searches. The first deep learning algorithms to do classification and regression of GWs emitted by non-spinning BBHs on quasi-circular orbits were presented in [13] in the context of simulated LIGO noise. The extension of that study to realistic detection scenarios using real advanced LIGO noise was introduced in [14]. Even though these algorithms were trained to do real-time classification and regression of GWs in realistic detection scenarios for a 2-D signal manifold (non-spinning BBHs on quasi-circular orbits), the studies presented in [13–16] have demonstrated that deep learning algorithms generalize to new types of sources, enabling the identification of moderately eccentric BBH mergers, spin precessing BBH mergers, and moderately eccentric BBH signals that include higher-order modes, respectively. These studies also indicate that while the detection of these new types of GW sources is possible, it is necessary to use higher-dimensional signal manifolds to train these algorithms to improve parameter estimation results, and to go beyond point-parameter estimation analysis. This work has

sparked the interest of the GW community, leading to a variety of studies including the classification of simulated BBH waveforms in Gaussian noise, GW source modeling and GW denoising of BBH mergers [15–26].

While detection and parameter estimation are the key goals for the development of deep learning for GW astrophysics, in this article we focus on the application of deep learning for parameter estimation. At present, GW parameter estimation is done using Bayesian inference [27–29], which is a well tested and extensively used method, though computationally-intensive. On the other hand, given the scalability of deep learning models in training mode (i.e., the ability to combine distributed training and large datasets to enhance the performance of deep learning algorithms in realistic data analysis scenarios), and their computational efficiency in inference mode, it is natural to explore their applicability for GW parameter estimation, the theme of this article.

Previous Work The first exploration of deep learning for the detection and point-parameter estimation of a 2-D signal manifold was presented in [13, 14]. For waveform signals with matched-filtering signal-to-noise ratio (SNR) SNR $\gtrsim$ 10, these neural network models measure the masses of quasi-circular BBH mergers with a mean percentage absolute error $\lesssim$ 15%, and with errors $\lesssim$ 35% for moderately eccentric BBH mergers. These results provided a glimpse of the robustness and scalability of deep neural network models, and the motivation to take these prototypical applications into a production run toolkit for GW parameter estimation.

Highlights of This Work

- We have designed new architectures and training schemes to demonstrate that deep learning provides the means to reconstruct the parameters of BBH mergers in more realistic astrophysical settings, i.e., BHs whose spins are aligned or anti-aligned, and which evolve on quasi-circular orbits. This 4-D signal manifold marks the first time deep learning models *at scale* are used for GW data analysis, i.e., models trained using datasets with tens of millions of waveforms, and 1,024 nodes (64 processor per node) to significantly reduce the training stage. Once fully trained, these deep learning models can reconstruct in real-time the parameters of the BBH catalog presented by the LIGO and Virgo Scientific Collaboration in [6].

- The neural network models we introduce in this article have two different architectures. The first one is tailored for the measurement of the masses of the binary components, whereas the second is used to quantify the final spin and the quasi-normal modes (QNMs) of the BH remnant. Once both neural networks are fully trained, we use them in parallel for inferences studies, finding that we can reconstruct the parameters of BBH mergers within 2 milliseconds using a single Tesla V100 GPU.

- We introduce a novel scheme to train Bayesian Neural Network (BNN) models at scale using 1,024 nodes on a High Performance Computing platform while keeping optimal performance for inference. We then adapted this framework to introduce for the first time the use of BNNs for GW parameter estimation. With this approach we can estimate the astrophysical parameters of the existing catalog of detected BBH mergers [6], and their posterior distributions, reporting inference times in the order of milliseconds.

- We use variational inference to approximate the posterior distribution of model parameters in the probabilistic layers of our neural networks. In the inference stage, we sample the network parameters to evaluate the posterior distribution of the physical parameters. Details of the model and training are in Sections II C and II F.

This article is structured as follows. Section II introduces the model architectures used in these analyses, it describes the construction and curation of the datasets used to train, validate and test our neural network models. It also includes a revised curriculum learning for neural network training. We quantify the accuracy of these neural network models in realistic detection scenarios using real advanced LIGO noise in Section III. We put at work our deep learning algorithms in Section IV to estimate the astrophysical parameters of the BBH mergers reported in [6]. We summarize our findings and future directions of work in Section V.

II. METHODS

In this section, we introduce the neural network models used for parameter estimation, and describe a novel curriculum learning scheme to accurately measure the masses of the binary components, and the final spin and QNMs of the BH remnant. We have used `TensorFlow` [30, 31] to design, train, validate and test the neural network models presented in this section.

The rationale to use two neural network models stems from the fact that the masses, spins and QNMs span rather different scales. Therefore, to improve the accuracy with which deep learning can measure these parameters we have designed one neural network that is tailored to measure the masses of the binary components, and one to measure the final spin and QNMs of the remnant. The astute reader may have noticed that the final spin of the BH remnant and its QNMs have a similar range of values when the QNMs are cast in dimensionless units, and this is the approach we have followed. In practice, we train the second neural network model using the fact that the QNMs are determined by the final spin a_f using the relation [32]

$$\omega_{220}(a_f) = \omega_R + i\,\omega_I\,, \tag{1}$$

where (ω_R, ω_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
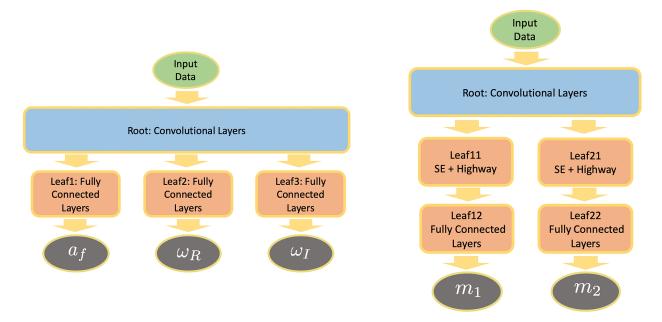
FIG. 1. The left architecture is used to estimate the final spin and quasi-normal modes of the black hole remnant. The right architecture is used to estimate the masses of the binary black hole components.

have computed the QNMs following [32]. One can readily translate ω_R into the ringdown frequency (in units of Hertz) and ω_I into the corresponding (inverse) damping time (in units of seconds) by computing $M_f \omega_{220}$. M_f represents the final mass of the remnant, and can be determined using Eq. (1) in [33].

As we describe below, we have found that to accurately reconstruct the masses of the binary components, it is necessary to use a more complex and deeper neural network architecture. It is worth mentioning that once these models are fully trained, a single GPU is sufficient to perform regression analyses in milliseconds using both neural network models.

A. Neural network model to measure the properties of the black hole remnant

The neural network model consists of two main parts: a shared root component for all physical parameters, and three leaf components for individual parameters (a_f, ω_R, and ω_I), as illustrated in the left panel of Figure 1, and Table I. The model architecture looks like a rooted tree. The root is composed of seven convolutional layers, and its output is shared by the leaves. Each leaf component has the same network architecture with three fully connected layers. This approach is inspired by the hierarchical self decomposing of convolutional neural networks described in [34, 35]. The key idea behind this approach is that the neural network structures are composed of a general feature extractor for the first seven layers, which is then followed up by sub-networks that take values from the output of the general feature extractor.

The rationale to have splits after the universal structure is to use sub-structures that focus on different subgroups of the data. As a simile: even though the human body has multiple limb locations ("leaves"), human motion is controlled by the overall motion of the body ("the root"). In practice this means that the tree structure of our models leverages the hierarchical structure of the data. It first extracts the universal features through the root, and then passes the information to the different sub-networks ("leaves") to learn specialized features for different physical parameters. Notice that the root will also prevent overfitting in the "leaves", since each leaf is optimized through the root.

Another change to the conventional architecture is that we remove the nonlinear activation in the second to last layer in the leaf component, i.e., it is a linear layer with identity activation function (see Table I). This allows more neurons to be activated and passed to the final layer. As discussed in [36], removing the nonlinear activation in some intermediate layers smooths the gradients and maintains the correlation of the gradients in the neural network weights, which, in turn, allows more information to be passed through the network as the depth increases.

B. Neural network model to measure the masses of the binary components

The tree-like network model used for this study is described in the right panel of Figure 1 and Table II. With respect to the architecture described in the previous section, we reduce the number of convolutional layers in the

TABLE I. Architecture of the neural network model used to measure the final spin and QNMs of the black hole remnant. For the root convolutional layers, the setup indicates: (kernel size, # of output channels, stride, dilation rate, max pooling kernel, max pooling stride). All convolutional layers have ReLU activation function and the padding is set to "VALID" mode. There is no max pooling layer if the last two entries in the configuration are 0's. The leaf fully connected layers setup: (# of output neurons, dropout rate). For the last layer, we use tanh activation function. However, the activation function in the second last layer is removed.

Layer Component	Layer Configurations	Activation Functions
Root Layer: Convolutional	$(16, 64, 1, 1, 4, 4)$ $(16, 128, 1, 2, 4, 4)$ $(16, 256, 1, 2, 4, 4)$ $(32, 256, 1, 2, 4, 4)$ $(4, 128, 1, 2, 0, 0)$ $(4, 128, 1, 2, 0, 0)$ $(2, 64, 1, 1, 0, 0)$	ReLU
Leaf Layer: Fully Connected	$(128, 0.0)$ $(128, 0.0)$ $(1, 0.0)$	ReLU Identity Tanh

root from seven to three. We have done this because we are now using more layers in the leaves, which in turn makes the gradient back-propagation harder. Reducing the number of root layers improves gradient updates to the front layers.

Each leaf component uses a squeeze-and-excitation (SE) structure [35]. The SE block is a sub-structure between two layers (squeeze step). It applies a global pooling, and assigns weights to each of the channels in the convolutional layers (excitation step). Compared to conventional convolutional structures with universal weights, the SE components adjust the importance of each channel with an adaptively learned weight, which, as described in [35], effectively results in 25% improvement in image classification. For images, channels are usually represented in RGB. Since we are using 1-D time-series signals, we treat channels of the original input signals to be 1. The SE block adaptively recalibrates channel-wise feature responses. Furthermore, the weights are optimally learned through a constraint introduced by the global pooling. This ensures that the weights encode both spatial and channel-wise information. Furthermore, the weights help the channels represent group specific features at deeper layers, which is consistent with our objective of using "leaves" for different parameters.

Following the SE components, the neural networks have two highway blocks [37]. The structures are a variant of the residual structure, as proposed in [38]. In the residual block, instead of directly learning the feature, it learns the residual components by an identity shortcut connection, which resolves the gradients vanishing when the model goes deeper. The highway block only introduces weights to the components in the residual block,

TABLE II. Architecture of the neural network model used to measure the masses of the binary components. For the root convolutional layers, the setup indicates: (kernel size, # of output channels, stride, dilation rate, max pooling kernel, max pooling stride). All convolutional layers have ReLU activation function and the padding is set to "VALID" mode. For the Leaf SE layer, the setup is: (# of output channels, # of residual blocks). The general structure for the SE layer follows the configuration described in [35]. Leaf highway layer setup: (kernel size, # of channels, stride, # of highway blocks). The configuration for the highway is described in [37]. The leaf fully connected layers setup is: (# of output neurons, dropout rate). For the last layer we use ReLU activation. However, the activation function in the second last layer is removed.

Layer Component	Layer Configurations	Activation Functions
Root Layer: Convolutional	$(16, 64, 1, 2, 4, 4)$ $(16, 128, 1, 2, 4, 4)$ $(16, 128, 1, 2, 4, 4)$	ReLU
Leaf Layer: SE	$(128, 3)$ $(128, 3)$	ReLU
Leaf Layer: Highway	$(4, 128, 2, 30)$	ReLU
Leaf Layer: Fully Connected	$(512, 0.1)$ $(256, 0.1)$ $(1, 0.0)$	ReLU Identity ReLU

which is similar to the application of importance weights on channels in SE components. Finally, we apply three fully connected layers with dropouts after the highway blocks to prevent overfitting [39]. The same nonlinearity reduction is also applied in the second last layer.

C. Probabilistic Model

In this section we present the probabilistic framework based on Bayesian inference, which we have applied to the neural networks outlined in Sections II B and II A. We use Bayesian neural networks (BNNs) [40, 41], which are neural networks with uncertainty over their weights, to provide estimates of the BBH masses and properties of the BH remnant posterior distributions. This is in contrast to standard neural networks which provide point estimates of parameters. We use prior and posterior distribution functions on the last two layers of each leaf. With this approach, each of the leaves becomes an independent probabilistic model that regresses the physical parameters. The root layers, on the other hand, can be viewed as feature extractors for each probabilistic leaf.

A BNN can be viewed as a probabilistic model for the posterior distribution, $p(\boldsymbol{w}|\mathcal{D})$, where $\boldsymbol{w}$ are the model weights and $\mathcal{D} = \{\boldsymbol{x}_j, \boldsymbol{y}_j\}_{j=1}^{n}$ is the training dataset. Here, $\boldsymbol{x}_j$ are the input noisy waveforms and $\boldsymbol{y}_j$ are the continuous parameters of interest, i.e., the BBH masses and the properties of the BH remnant.

According to Bayes theorem, $p(\boldsymbol{w}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{w})p(\boldsymbol{w})$,

where $p(\boldsymbol{w})$ is the prior distribution for the weights and $p(\mathcal{D}|\boldsymbol{w})$ is the likelihood. We assume that the likelihood function for each pair of the training data is,

$$p\left(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w}\right) = \frac{1}{\sqrt{2\pi}\epsilon} \exp\left(-\frac{\|\boldsymbol{y} - f_{\boldsymbol{w}}(\boldsymbol{x})\|^2}{2\epsilon^2}\right), \quad (2)$$

where $f_{\boldsymbol{w}}$ represents the neural network function with weights $\boldsymbol{w}$ and ϵ is the standard deviation. The aleatoric uncertainty is covered by the likelihood distribution. A BNN allows a stochastic sampling of the weight parameters during a forward pass through the network while also encoding prior knowledge through the use of prior distributions. We use a variational inference (VI) algorithm to approximate the weight posterior distribution $p(\boldsymbol{w}|\mathcal{D})$ using a Gaussian distribution for the weights assuming a mean field approximation, denoted by $q_{\boldsymbol{\theta}}(\boldsymbol{w})$. It is parameterized by $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$, representing the mean vector and the standard deviation vector of the distribution respectively.

The corresponding cost function can be written as

$$\mathcal{L} = \mathrm{KL}\left(q_{\boldsymbol{\theta}}(\boldsymbol{w})\|p(\boldsymbol{w})\right) - \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{w})} \log p(\mathcal{D}|\boldsymbol{w}), \quad (3)$$

which is known as the variational free energy. The prior distribution is chosen to be a standard normal distribution. Since the probabilistic layers are parameterized by the mean and variance of the weight distributions, the number of parameters which need to be optimized is doubled compared to a standard neural network. The cost function can be approximated by drawing N samples $\boldsymbol{w}^{(i)}$ from $q_{\boldsymbol{\theta}}(\boldsymbol{w})$,

$$\mathcal{L} \approx \frac{1}{N} \sum_{i=1}^{N} \left[-\log q_{\boldsymbol{\theta}}\left(\boldsymbol{w}\right) - \log p\left(\boldsymbol{w}^{(i)}\right) \right.$$
$$\left. - \log p\left(\mathcal{D}|\boldsymbol{w}^{(i)}\right) \right] \quad (4)$$

During training, for every forward model pass, the variational posterior distribution for the model parameters is estimated. Specifically, we use stochastic gradient descent to estimate $\boldsymbol{\theta}$ of $q_{\boldsymbol{\theta}}(\boldsymbol{w})$ by minimizing Eq. (4). In testing or inference mode, for input waveform $\boldsymbol{x}^*$, our approximate predictive distribution is given by,

$$q(\boldsymbol{y}^*|\boldsymbol{x}^*) = \int p(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{w}) q_{\boldsymbol{\theta}}(\boldsymbol{w}) \, d\boldsymbol{w}. \quad (5)$$

We use sampling to compute the statistics of the corresponding estimated physical parameters, e.g., median and 90% confidence interval. In addition to the aleatoric uncertainty, the uncertainty in the predictions arises from uncertainty in the weights or so called 'epistemic uncertainty.'

In this probabilistic modeling, we apply the following simplifications: (1) the likelihood function is assumed to be Gaussian, and (2) neural network weight distributions are assumed to be independent Gaussians. Under these assumptions, the loss in Eq. (4) is simplified and tractable. The statistical models and VI method are implemented using the computing framework TensorFlow Probability (TFP) [42, 43] using a modified sampling scheme and distributed across nodes in a data parallel fashion using Horovod [44]. Details of the model training at scale are discussed in Section II F.

D. Dataset Preparation

To demonstrate the use of deep learning for parameter estimation, we consider the catalog of BBH mergers presented in [6]. Based on the Bayesian analyses presented in that study, we consider the following parameter space to produce our training dataset: $m_1 \in [9\mathrm{M}_\odot, 65\mathrm{M}_\odot]$, $m_2 \in [5.2\mathrm{M}_\odot, 42\mathrm{M}_\odot]$. The spin of the binary components span a range $a_{\{1,2\}} \in [-0.8, 0.8]$. By uniformly sampling this parameter space we produce a dataset with 300,180 waveforms. These waveforms are produced with the surrogate waveform family [45], considering the last second of the evolution which includes the late inspiral, merger and ringdown. The waveforms are produced using a sample rate of 8192Hz.

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 BH remnant using the formulae provided in [46], and the QNMs of the ringdown following [32]. In essence, we are training our neural network models to identify the key features that determine the properties of the BBHs before and after merger using a unified framework.

In order to encapsulate the true properties of advanced LIGO noise, we whiten all the training templates using real LIGO noise from the Hanford and Livingstone detectors gathered during the first and second observing runs [47].

We use 70% of these waveform samples for training, 15% for validation, and 15% for testing. The training samples are randomly and uniformly chosen. Throughout the training, we use ADAM optimizer to minimize the mean squared error of the predicted parameters with default hyper-parameter setups [48]. We choose the batch size to be 64, the learning rate to be 0.0008, the total number of iterations to be 120,000 (maximum). We use a dropout rate 0.1 for training and no dropout is applied for testing and validation. To simulate the environment where the true GWs are embedded, we use real advanced LIGO noise to compute power spectral density, which is then used to whiten the templates. In addition, we apply a random 0% to 6% left or right shifts. This endows the neural networks with time-invariance, and improves their performance to estimate the parameters

TABLE III. Decreasing peak SNR (pSNR) setup. The pSNR is uniformly chosen within the indicated range. Notice that the early stopping criterion is also applied if the number of iterations is greater than 60,000 and the relative error threshold is met. The relation between match-filtering SNR to pSNR is: 1.0 pSNR ≈ 13.0 SNR.

Iterations	pSNRs
1-12000	2.0-3.0
12001-24000	1.5-3.0
24001-36000	1.0-3.0
36001-60000	0.5-3.0
60001-90000	0.3-3.0
90001-120000	0.2-3.0
120001-	0.1-3.0

of the signal irrespective of their position in the data stream. On the other hand, this technique also prevents overfitting of the data. Since the locations are randomly shifted with independent noise injected, the training data are different at each epoch.

E. Curriculum learning with decreasing signal-to-noise ratio

In realistic detection scenarios, GWs have moderate SNRs, and are contaminated by non-Gaussian and non-stationary noise. In order to ensure that neural networks identify GWs over a broad range of astrophysically motivated SNRs, we start training them with large SNRs, and gradually reduce the SNRs to a lower level. This is an idea taken from curriculum learning literature [49], which allows the network to distill more accurate information of the underlying signals with larger SNRs to signals with lower SNRs. This approach has been demonstrated for classification, regression and denoising of GW signals [13–17, 19, 50, 51]. Specifically, each waveform is normalized to have maximum amplitude 1, and then we use curriculum learning with the decreasing SNR scheme detailed in Table III (The strategy for BNN models is the same). The noisy data is then normalized to have variance one. We normalize the data to ensure that the trained model can characterize true BBH signals in realistic detection scenarios, covering a broad range of SNRs.

The different steps followed in our curriculum learning scheme are presented in Table III. In addition, we use an early stopping criterion with the relative error threshold 0.026 for (m_1, m_2) and 0.0016 for $(a_f, \omega_R, \omega_I)$. One additional change to the mass model is we rescale the masses by 1/20, to make the optimization converge faster. In the evaluation, we just scale the data back to its original amplitude.

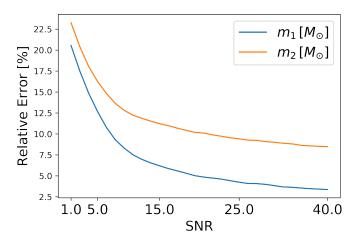### F. Training of the Bayesian Neural Network Model

For the probabilistic layers, as the effective number of parameters to be optimized is double that of a standard layers, we examine the impact of scaling the BNN code across nodes on the pre-exascale Cray XC40 system, Theta, at Argonne National Laboratory. Using an optimized build of both Tensorflow and Horovod for the Intel Xeon-Phi [coded name Knights Landing (KNL)] architecture we distribute the code using one MPI rank per node and 128 hardware threads per node and scale up to 1024 nodes. Results for the number of samples processed per second during training is shown in Figure 3. We achieve ∼ 75% efficiency up to 1024 nodes on Theta. As the number of nodes is increased, there is increased communication of the gradients at each iteration which causes an expected decrease in performance away from the ideal scaling. As the BNN layers have in effect twice the parameters of the standard layers, the communication cost is slightly higher which can be seen as a decrease in the number of samples processed per second.

In addition to evaluating the efficiency on Theta, we fully trained the two BNN models on Hardware-Accelerated Learning (HAL) cluster at the National Center for Supercomputing Applications. Each model was trained on 4 NVIDIA V100 GPUs with batch size of 64. The parameter ϵ in the likelihood function Eq. (2) is chosen to be 0.1 for the mass model and 10^{-3} for the final spin and QNMs model. We draw $N = 100$ samples and $M = 1600$ samples from $q_{\boldsymbol{\theta}}(\boldsymbol{w})$ at training and testing respectively. The learning rate for the two BNN models is 8×10^{-6}. The total number of iterations is 200,000 to guarantee convergence.

III. EXPERIMENTAL RESULTS

Using the signal manifold described in the previous section, we present results of the accuracy with which our neural network models can measure the masses of the binary components, and the properties of the corresponding remnant.

Figure 2 presents the accuracy with which the binary components (m_1, m_2) can be recovered over a a broad range of SNRs. We notice that for signals with SNR ≥ 15, the primary and secondary masses can be constrained with relative errors [52] less than (7%, 12%), respectively. These results represent a major improvement to the analysis we reported in the context of a 2-D signal manifold in [13, 14]. Furthermore, we can also see from the same figure that for signals with SNR ≥ 15 our neural network models can measure the triplet $(a_f, \omega_R, \omega_I)$ with relative errors less than (13%, 5%, 3%), respectively. To the best of our knowledge, this is the first time deep learning is used to infer the properties of BH remnants directly from GW signals.
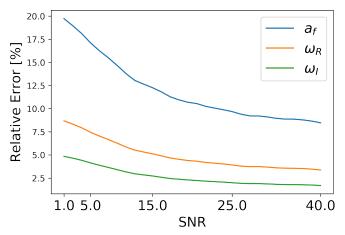
FIG. 2. Relative error with which our deep learning algorithm can measure the masses, final spin, a_f, and quasi-normal modes (QNMs), (ω_R, ω_I) of the binary black hole components as a function of optimal matched-filtering signal-to-noise ration (SNR). *Left panel:* For waveform with SNR $\geq$ 15, the primary and secondary masses can be constrained with relative errors less than (7%, 12%), respectively. *Right panel:* For signals with SNR $\geq$ 15, $(a_f, \omega_R, \omega_I)$ can be recovered with relative errors less than (13%, 5%, 3%), respectively.

FIG. 3. Samples processed per second with increasing number of nodes during training of the neural network. The results for the BNN are shown in cyan and standard neural network in blue. Ideal scaling is shown as a dashed black bar at each node count. Error bars are the variance from all iterations during training.

IV. DEEP LEARNING PARAMETER ESTIMATION OF DETECTED BINARY BLACK HOLE MERGERS

In this section we use our neural network models to measure $(m_1, m_2, a_f, \omega_R, \omega_I)$ from all the BBH mergers detected to date by the advanced LIGO and Virgo observatories [6]. We present results for two types of neural network models, namely, deterministic and probabilistic.

A. Parameter estimation with deterministic neural networks

To get insights into the performance of our deterministic neural network models to infer the astrophysical parameters of BBH mergers, we begin by evaluating them for a given BBH system whose ground truth parameters are $(m_1, m_2, a_f, \omega_R, \omega_I) = (31.10 M_\odot, 20.46 M_\odot, 0.718, 0.5412, 0.0800)$. Using 1,600 different noise realizations, we have constructed the model predictions for two different SNR cases, as shown in Figure 4. We notice that these distributions capture the ground-truth values of the BBH system under consideration, and that the reconstruction of the actual parameters of the system improves for larger SNR values, which is in agreement with the analysis presented with traditional Bayesian analysis for GW parameter estimation [28]. Having conducted similar experiments for other BBH systems, we then went on to using these deep learning models for the parameter reconstruction of real BBH mergers.

In Table IV we present the median and 90% confidence level for the astrophysical parameters $(m_1, m_2, a_f, \omega_R, \omega_I)$ of all the BBH mergers presented in [6]. These values are computed by whitening the data containing a putative signal with 240 different Power Spectral Densities (PSDs), half of them are constructed using LIGO Hanford noise and the rest with LIGO Livingstone noise. Through this approach we are effectively measuring the impact of PSD variations in the measurements of the astrophysical parameters of BBH mergers. We find that these estimates are in very good agreement with the results obtained with the Bayesian analyses presented in Table III of [6].

TABLE IV. Parameter estimation results for the catalog of binary black hole mergers reported in [6] using our deterministic deep learning models. We report median values with the the 90% confidence interval, which was computed by whitening gravitational wave strain data that contain real gravitational wave signals with up to 240 different power spectral densities.

Event Name	$m_1\,[\mathrm{M_\odot}]$	$m_2\,[\mathrm{M_\odot}]$	a_f	ω_R	ω_I
GW150914	$35.64^{+5.19}_{-5.55}$	$29.74^{+2.12}_{-3.90}$	$0.658^{+0.039}_{-0.006}$	$0.5253^{+0.0186}_{-0.0026}$	$0.0820^{+0.0002}_{-0.0009}$
GW151012	$25.01^{+12.00}_{-9.09}$	$16.45^{+4.50}_{-6.01}$	$0.637^{+0.011}_{-0.015}$	$0.5155^{+0.0028}_{-0.0086}$	$0.0824^{+0.0002}_{-0.0002}$
GW151226	$12.39^{+3.57}_{-0.25}$	$7.70^{+5.77}_{-0.48}$	$0.725^{+0.051}_{-0.140}$	$0.5558^{+0.0241}_{-0.0611}$	$0.0776^{+0.0055}_{-0.0002}$
GW170104	$32.28^{+4.31}_{-6.33}$	$22.31^{+7.01}_{-3.06}$	$0.684^{+0.014}_{-0.035}$	$0.5157^{+0.0071}_{-0.0068}$	$0.0854^{+0.0004}_{-0.0015}$
GW170608	$12.90^{+3.27}_{-0.31}$	$9.93^{+2.08}_{-0.09}$	$0.716^{+0.017}_{-0.077}$	$0.5385^{+0.0057}_{-0.0154}$	$0.0827^{+0.0006}_{-0.0004}$
GW170729	$45.32^{+2.23}_{-0.98}$	$24.41^{+03.16}_{-02.32}$	$0.737^{+0.036}_{-0.058}$	$0.5682^{+0.0038}_{-0.0303}$	$0.0739^{+0.0054}_{-0.0016}$
GW170809	$35.71^{+7.53}_{-8.46}$	$24.09^{+5.80}_{-2.44}$	$0.632^{+0.008}_{-0.010}$	$0.5123^{+0.0034}_{-0.0041}$	$0.0826^{+0.0001}_{-0.0002}$
GW170814	$30.54^{+2.01}_{-8.78}$	$22.33^{+0.07}_{-7.96}$	$0.679^{+0.002}_{-0.003}$	$0.5364^{+0.0009}_{-0.0030}$	$0.0812^{+0.0003}_{-0.0001}$
GW170818	$31.52^{+2.15}_{-1.95}$	$25.97^{+1.21}_{-0.87}$	$0.716^{+0.015}_{-0.021}$	$0.5474^{+0.0062}_{-0.0104}$	$0.0786^{+0.0013}_{-0.0013}$
GW170823	$46.98^{+0.58}_{-3.89}$	$33.01^{+2.03}_{-5.92}$	$0.626^{+0.014}_{-0.023}$	$0.5067^{+0.0070}_{-0.0057}$	$0.0827^{+0.0006}_{-0.0003}$

TABLE V. As Table IV, but now using our probabilistic deep learning models. The uncertainty for these models is captured by randomness in the network weights, not from various noisy realizations of the signals.

Event Name	$m_1[\mathrm{M_\odot}]$	$m_2[\mathrm{M_\odot}]$	a_f	ω_R	ω_I
GW150914	$36.08^{+4.77}_{-4.45}$	$27.42^{+3.49}_{-3.92}$	$0.689^{+0.017}_{-0.032}$	$0.5390^{+0.0124}_{-0.0269}$	$0.0797^{+0.0011}_{-0.0022}$
GW151012	$21.56^{+3.07}_{-2.12}$	$15.46^{+2.44}_{-2.32}$	$0.681^{+0.016}_{-0.032}$	$0.5365^{+0.0130}_{-0.0266}$	$0.0804^{+0.0008}_{-0.0018}$
GW151226	$18.04^{+1.98}_{-2.49}$	$11.96^{+1.67}_{-2.89}$	$0.715^{+0.017}_{-0.035}$	$0.5533^{+0.0142}_{-0.0280}$	$0.0763^{+0.0017}_{-0.0036}$
GW170104	$31.23^{+4.09}_{-3.26}$	$23.27^{+3.62}_{-3.25}$	$0.692^{+0.016}_{-0.033}$	$0.5358^{+0.0052}_{-0.0302}$	$0.0796^{+0.0026}_{-0.0052}$
GW170608	$16.73^{+2.38}_{-2.19}$	$12.44^{+2.03}_{-2.21}$	$0.673^{+0.019}_{-0.036}$	$0.5235^{+0.0149}_{-0.0297}$	$0.0818^{+0.0005}_{-0.0012}$
GW170729	$45.28^{+6.63}_{-6.42}$	$32.34^{+3.91}_{-5.48}$	$0.751^{+0.019}_{-0.038}$	$0.5776^{+0.0151}_{-0.0309}$	$0.0756^{+0.0023}_{-0.0048}$
GW170809	$32.88^{+4.49}_{-3.45}$	$26.56^{+3.54}_{-3.91}$	$0.714^{+0.016}_{-0.034}$	$0.5492^{+0.0060}_{-0.0271}$	$0.0760^{+0.0030}_{-0.0060}$
GW170814	$32.40^{+4.77}_{-3.60}$	$25.22^{+4.38}_{-4.34}$	$0.675^{+0.016}_{-0.033}$	$0.5329^{+0.0140}_{-0.0272}$	$0.0794^{+0.0011}_{-0.0024}$
GW170818	$33.49^{+4.51}_{-3.19}$	$29.71^{+4.59}_{-4.63}$	$0.631^{+0.015}_{-0.032}$	$0.5159^{+0.0135}_{-0.0277}$	$0.0829^{+0.0008}_{-0.0016}$
GW170823	$38.24^{+4.78}_{-5.38}$	$28.16^{+4.63}_{-3.67}$	$0.664^{+0.018}_{-0.036}$	$0.5321^{+0.0156}_{-0.0302}$	$0.0757^{+0.0043}_{-0.0088}$

B. Bayesian neural network parameter estimation

In addition to parameter estimation results obtained with our deterministic models, based on varying the noise realization with different PSDs, we also evaluated our BNN models for two types of signals. First, on simulated signals to quantify the performance of our probabilistic models. Results of this exercise are presented in Figure 5. We carried out an exhaustive study to confirm that our BNN models provide consistent results for different random initializations, and that the results exhibit strong convergence for the optimal choice of hyperparameters.
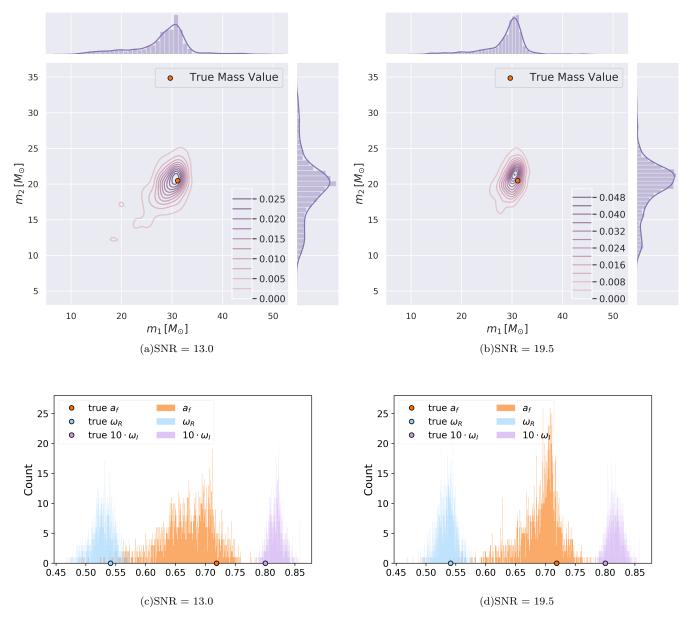
FIG. 4. Model predictions produced by our deterministic models by evaluating them with 1,600 different noise realizations for a binary black hole system with ground truth parameters $(m_1, m_2, a_f, \omega_R, \omega_I) = (31.10 M_\odot, 20.46 M_\odot, 0.718, 0.5412, 0.0800)$. The panels show results for the distribution of the estimates for $(m_1, m_2, a_f, w_R, w_I)$ assuming SNR = {13, 19.5}.

Upon confirming that our probabilistic models perform well, we used them to estimate the astrophysical parameters of the entire catalog of BBH signals reported in [6]. These results, which provide the median and the 90% confidence intervals, are summarized in Table V.

The deep learning parameter estimation results presented in Tables V are consistent with those obtained with established, Bayesian parameter estimation pipelines [6]. The reliable astrophysical information inferred in low-latency by these deep learning algorithms for each BBH signal (less than 2 milliseconds) warrants the extension of this framework to characterize other

GW sources, including eccentric compact binary mergers, and sources such as BBH systems with significant spin and asymmetric mass-ratios that require the inclusion of higher-order modes for accurate GW source modeling. This work is under earnest development and will be presented shortly.

Having demonstrated the application of deep learning at scale for the characterization of BBH mergers, it is now in order to design deep neural networks for real-time detection and characterization of GW sources that are expected to have electromagnetic and astro-particle counterparts, i.e., BNS and NSBH systems. For that

(a)SNR = 13.0

(b)SNR = 19.5

(c)SNR = 13.0

(d)SNR = 19.5

FIG. 5. Variation inference distributions produced by our Bayesian Neural Network models for a binary black hole system with ground truth parameters $(m_1, m_2, a_f, \omega_R, \omega_I) = (31.10 M_\odot, 20.46 M_\odot, 0.718, 0.5412, 0.0800)$. The panels show results for the distribution of the estimates for $(m_1, m_2, a_f, w_R, w_I)$. As in Figure 4, we consider SNR = {13, 19.5}.

study, we expect no additional computational challenges to the ones we have already addressed in this analysis. The central development for such an effort, however, will consist of designing a clever algorithm to readily identify BNS or NSBH in a hierarchical manner, i.e., in principle it is not needed to train neural networks using minute long waveforms. Rather, we need to figure out how much information is needed to accurately reconstruct the astrophysical parameters of one of these events in real-time. These studies should be pursued in the future.

V. CONCLUSION

We have presented the first application of deep learning at scale to characterize the astrophysical properties of BHs whose spins are aligned or anti-aligned, and which evolve on quasi-circular orbits. Using over 10^7 waveforms to densely sample this parameter space, and encoding time- and scale-invariance, we have demonstrated that deep learning enables real-time GW parameter estimation. These studies mark the first time BNNs are trained

using 1,024 nodes on a supercomputer platform tuned for deep learning research, and when applied for the analysis of real advanced LIGO data, they maintain similar accuracy to models trained on 4 V100 GPUs. Our results are consistent with established, compute-intensive, Bayesian methods that are routinely used for GW parameter estimation.

The approach we have presented herein provides the means to constrain the parameters of BBHs before and after the merger event. We have shown that deep learning can directly infer the final spin and QNMs of BH remnants, thereby paving the way to directly use QNMs to assess whether BH remnants are accurately described by general relativity. In future work, we will study how accurately these neural network models can tell apart ringdown waveforms described by astrophysically motivated alternative theories of gravity in realistic detection scenarios. The extension of this work to enable real-time detection and parameter estimation of GW sources that are central for Multi-Messenger Astrophysics discovery campaigns, and other astrophysically motivated sources, such as eccentric BBH mergers, should also be investigated.

VII. CONTRIBUTION

EAH envisioned this study, and directed the construction of the data sets used to train/validate/test the neural network models. H Shen developed the neural network structure and carried out the training and evaluation. ZZ supervised on the evaluation of the neural network performance. EJ created the BNN, implemented this to run at scale and advised on its use for parameter predictions. H Shen created the BNN code with a new sampling approach and training objective, trained and evaluated the BNN model on HAL machine for parameter predictions on real GW events. H Sharma developed the BNN code and carried out extensive scaling tests on Theta. All co-authors contributed to drafting and editing the manuscript.

[1] B. P. Abbott, et al., Physical Review Letters 116, 061102 (2016), arXiv:1602.03837 [gr-qc].
[2] B. P. Abbott, et al., Physical Review Letters 116, 241103 (2016), arXiv:1606.04855 [gr-qc].
[3] B. P. Abbott, et al., Physical Review Letters 118, 221101 (2017).
[4] B. P. Abbott, et al., Physical Review Letters 119, 141101 (2017), arXiv:1709.09660 [gr-qc].
[5] B. P. Abbott, et al., Astrophys. J. Lett 851, L35 (2017), arXiv:1711.05578 [astro-ph.HE].
[6] B. P. Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration), Phys. Rev. X 9, 031040 (2019).
[7] B. P. Abbott, et al., Physical Review Letters 116, 131103 (2016), arXiv:1602.03838 [gr-qc].
[8] The LIGO Scientific Collaboration, J. Aasi, et al., Classical and Quantum Gravity 32, 074001 (2015), arXiv:1411.4547 [gr-qc].
[9] F. Acernese et al., Classical and Quantum Gravity 32, 024001 (2015), arXiv:1408.3978 [gr-qc].
[10] LSST Science Collaboration, P. A. Abell, J. Allison, S. F. Anderson, J. R. Andrew, J. R. P. Angel, L. Armus, D. Ar-

nett, S. J. Asztalos, T. S. Axelrod, and et al., ArXiv e-prints (2009), arXiv:0912.0201 [astro-ph.IM].

[11] G. Allen, W. Anderson, E. Blaufuss, J. S. Bloom, P. Brady, S. Burke-Spolaor, S. B. Cenko, A. Connolly, P. Couvares, D. Fox, A. Gal-Yam, S. Gezari, A. Goodman, D. Grant, P. Groot, J. Guillochon, C. Hanna, D. W. Hogg, K. Holley-Bockelmann, D. A. Howell, D. Kaplan, E. Katsavounidis, M. Kowalski, L. Lehner, D. Muthukrishna, G. Narayan, J. E. G. Peek, A. Saha, P. Shawhan, and I. Taboada, arXiv e-prints , arXiv:1807.04780 (2018), arXiv:1807.04780 [astro-ph.IM].

[12] G. Allen, I. Andreoni, E. Bachelet, G. B. Berriman, F. B. Bianco, R. Biswas, M. Carrasco Kind, K. Chard, M. Cho, P. S. Cowperthwaite, Z. B. Etienne, D. George, T. Gibbs, M. Graham, W. Gropp, A. Gupta, R. Haas, E. A. Huerta, E. Jennings, D. S. Katz, A. Khan, V. Kindratenko, W. T. C. Kramer, X. Liu, A. Mahabal, K. McHenry, J. M. Miller, M. S. Neubauer, S. Oberlin, J. Olivas, Alexander R., S. Rosofsky, M. Ruiz, A. Saxton, B. Schutz, A. Schwing, E. Seidel, S. L. Shapiro, H. Shen, Y. Shen, B. M. Sipőcz, L. Sun, J. Towns, A. Tsokaros, W. Wei, J. Wells, T. J. Williams, J. Xiong, and Z. Zhao, arXiv e-prints , arXiv:1902.00522 (2019), arXiv:1902.00522 [astro-ph.IM].

[13] D. George and E. A. Huerta, Phys. Rev. D 97, 044039 (2018), arXiv:1701.00008 [astro-ph.IM].

[14] D. George and E. A. Huerta, Physics Letters B 778, 64 (2018), arXiv:1711.03121 [gr-qc].

[15] D. George and E. A. Huerta, in NiPS Summer School 2017 Gubbio, Perugia, Italy, June 30-July 3, 2017 (2017) arXiv:1711.07966 [gr-qc].

[16] A. Rebei, E. A. Huerta, S. Wang, S. Habib, R. Haas, D. Johnson, and D. George, Phys. Rev. D100, 044025 (2019), arXiv:1807.09787 [gr-qc].

[17] H. Shen, D. George, E. A. Huerta, and Z. Zhao, in ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2019) pp. 3237–3241.

[18] E. A. Huerta, D. George, Z. Zhao, and G. Allen, arXiv e-prints , arXiv:1805.02716 (2018), arXiv:1805.02716 [cs.LG].

[19] W. Wei and E. A. Huerta, arXiv e-prints , arXiv:1901.00869 (2019), arXiv:1901.00869 [gr-qc].

[20] A. J. K. Chua, C. R. Galley, and M. Vallisneri, Phys. Rev. Lett. 122, 211101 (2019), arXiv:1811.05491 [astro-ph.IM].

[21] H. Gabbard, M. Williams, F. Hayes, and C. Messenger, Physical Review Letters 120, 141103 (2018), arXiv:1712.06041 [astro-ph.IM].

[22] X. Fan, J. Li, X. Li, Y. Zhong, and J. Cao, ArXiv e-prints (2018), arXiv:1811.01380 [astro-ph.IM].

[23] J. A. González and F. S. Guzmán, Phys. Rev. D 97, 063001 (2018), arXiv:1803.06060 [astro-ph.HE].

[24] Y. Fujimoto, K. Fukushima, and K. Murase, Phys. Rev. D 98, 023019 (2018), arXiv:1711.06748 [nucl-th].

[25] X. Li, W. Yu, and X. Fan, ArXiv e-prints (2017), arXiv:1712.00356 [astro-ph.IM].

[26] H. Nakano, T. Narikawa, K.-i. Oohara, K. Sakai, H.-a. Shinkai, H. Takahashi, T. Tanaka, N. Uchikata, S. Yamamoto, and T. S. Yamamoto, Phys. Rev. D99, 124032 (2019), arXiv:1811.06443 [gr-qc].

[27] P. Graff, F. Feroz, M. P. Hobson, and A. Lasenby, MNRAS 421, 169 (2012), arXiv:1110.2997 [astro-ph.IM].

[28] J. Veitch, V. Raymond, B. Farr, W. Farr, P. Graff, S. Vitale, B. Aylott, K. Blackburn, N. Christensen, M. Coughlin, W. Del Pozzo, F. Feroz, J. Gair, C.-J. Haster, V. Kalogera, T. Littenberg, I. Mandel, R. O'Shaughnessy, M. Pitkin, C. Rodriguez, C. Röver, T. Sidery, R. Smith, M. Van Der Sluys, A. Vecchio, W. Vousden, and L. Wade, Phys. Rev. D 91, 042003 (2015), arXiv:1409.7215 [gr-qc].

[29] L. P. Singer and L. R. Price, Physical Review D 93, 024013 (2016), arXiv:1508.03634 [gr-qc].

[30] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., in 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16) (2016) pp. 265–283.

[31] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., Software available from tensorflow. org 1 (2015).

[32] E. Berti, V. Cardoso, and C. M. Will, Phys. Rev. D 73, 064030 (2006), arXiv:gr-qc/0512160.

[33] J. Healy and C. O. Lousto, Phys. Rev. D 95, 024037 (2017), arXiv:1610.09713 [gr-qc].

[34] K. Sairam, J. Mukherjee, A. Patra, and P. P. Das, CoRR (2018).

[35] J. Hu, L. Shen, and G. Sun, in Proceedings of the IEEE conference on computer vision and pattern recognition (2018) pp. 7132–7141.

[36] X. Dong, G. Kang, K. Zhan, and Y. Yang, CoRR (2017), arXiv:1709.07634.

[37] R. K. Srivastava, K. Greff, and J. Schmidhuber, arXiv e-prints , arXiv:1505.00387 (2015), arXiv:1505.00387 [cs.LG].

[38] K. He, X. Zhang, S. Ren, and J. Sun, in Proceedings of the IEEE conference on computer vision and pattern recognition (2016) pp. 770–778.

[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Journal of Machine Learning Research 15, 1929 (2014).

[40] R. M. Neal, Bayesian learning for neural networks, Vol. 118 (Springer Science & Business Media, 2012).

[41] D. J. MacKay, Neural computation 4, 448 (1992).

[42] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous, arXiv e-prints , arXiv:1711.10604 (2017), arXiv:1711.10604 [cs.LG].

[43] D. Tran, M. W. Dusenberry, M. van der Wilk, and D. Hafner, arXiv e-prints , arXiv:1812.03973 (2018), arXiv:1812.03973 [cs.LG].

[44] A. Sergeev and M. Del Balso, ArXiv e-prints (2018), arXiv:1802.05799.

[45] J. Blackman, S. E. Field, C. R. Galley, B. Szilágyi, M. A. Scheel, M. Tiglio, and D. A. Hemberger, Physical Review Letters 115, 121102 (2015), arXiv:1502.07758 [gr-qc].

[46] F. Hofmann, E. Barausse, and L. Rezzolla, Astrophys. J. 825, L19 (2016), arXiv:1605.01938 [gr-qc].

[47] LIGO Lab, the LIGO Scientific Collaboration and the Virgo Scientific Collaboration, "Gravitational wave open science center," (2019), https://www.gw-openscience.org/about/.

[48] D. P. Kingma and J. Ba, CoRR abs/1412.6980 (2014).

[49] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, in Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09 (ACM, New York, NY, USA, 2009) pp. 41–48.

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

[51] D. George, H. Shen, and E. A. Huerta, in *NiPS Summer School 2017 Gubbio, Perugia, Italy, June 30-July 3, 2017*

(2017) arXiv:1711.07468 [astro-ph.IM].

[52] M. Abramowitz and I. A. Stegun, *Dover Books on Advanced Mathematics, New York: Dover, —c1965, Corrected edition, edited by Abramowitz, Milton; Stegun, Irene A.* (1965).