

# Detection of Gravitational Waves Pipeline using Deep Learning

Malin Ouyang<sup>1, \*</sup>

<sup>1</sup>*The School of Astronomy and Space Science, Nanjing University, China*

Gravitational wave (GW) detection has achieved remarkable success, particularly in identifying signals from compact binary mergers, including binary black hole (BBH) and binary neutron star (BNS) systems. BNS mergers are of particular interest for multi-messenger astronomy due to their potential to produce electromagnetic counterparts. Recent advancements in deep learning have provided alternative approaches to traditional matched-filtering techniques, yet achieving sensitive and robust detection of GW signals from both BBH and BNS mergers remains challenging. In this study, we developed a deep learning framework that utilizes convolutional neural networks (CNNs) and ResNet architectures to detect GW signals embedded in Gaussian noise. We trained our models using 5,000 signal-plus-noise samples and 5,000 pure noise samples, with optimal signal-to-noise ratios (SNRs) of 20 for BBH and 30 for BNS signals. Receiver operating characteristic (ROC) and sensitivity curves were employed to evaluate model performance, revealing that the CNN model achieves high sensitivity for BBH signals but struggles with BNS detection due to dataset limitations. Incorporating on-the-fly data augmentation significantly enhanced training efficiency and robustness compared to traditional pre-computed augmentation methods, which, while more robust, required substantially greater computational resources. Model comparisons between CNN, ResNet50, ResNet101 and ResNet151 show that ResNet-based architectures, especially ResNet101, outperform CNNs in GW detection tasks, providing superior sensitivity at lower false-alarm probabilities. However, this comes with higher computational complexity, underscoring the trade-off between accuracy and efficiency. While our pipeline demonstrated competitive performance in detecting BBH signals, further optimization is required for BNS detection.

## I. INTRODUCTION

Gravitational waves (GW) from compact binary coalescences (CBCs) are now routinely detected by ground-based laser interferometers. The LIGO [1] and Virgo [2] observatories have identified over 90 CBCs during their first three observing runs [3–5]. Most of these events are binary black hole (BBH) mergers, with only two confirmed binary neutron star (BNS) mergers [6, 7], and two confirmed neutron star–black hole (NSBH) mergers reported by the end of the third observing run (O3) of the LIGO-Virgo-KAGRA collaboration. The first detected BNS merger, GW170817, marked the beginning of a new era of multi-messenger astronomy, with gravitational waves serving as a crucial messenger [6, 8]. A gamma-ray burst was serendipitously detected from this merger [9], along with a kilonova and X-ray counterpart identified through follow-up observations [8, 10]. These observations enabled unique measurements of the Hubble constant [11] and provided constraints on the neutron star equation of state [12, 13]. Further observations of BNS mergers could refine constraints on the Hubble constant, address the Hubble tension, and potentially reveal links between BNS mergers and other transient signals, such as fast radio bursts. As interferometer sensitivity improves and new instruments like KAGRA [14] come online, the increasing likelihood of multi-messenger detections highlights the need for developing new CBC search pipelines.

CBCs are currently detected using five primary search pipelines [15–19], four of which employ matched filtering to identify signals. These pipelines use a bank of signal templates with unique intrinsic parameters to cover the mass-spin parameter space. The templates are cross-correlated with incoming GW detector data to produce signal-to-noise ratio (SNR) time series. In the absence of noise, the highest SNR is achieved by the template whose parameters closely match those of the true signal. Triggers are generated when an SNR threshold is met (e.g.,  $\text{SNR} > 4$  in one detector). These triggers are then clustered and assigned a significance using a ranking statistic, which typically accounts for the peak SNR, coincident triggers between observing interferometers, and signal consistency tests [20]. Triggers are further evaluated by assigning a false alarm rate (FAR) based on background triggers, and those with sufficiently low FAR are considered GW candidates.

Despite the success of current pipelines in detecting CBCs, exploring new detection methods is worthwhile for several reasons. Firstly, the overall search for CBCs benefits from incorporating multiple pipelines with unique search methods [21]. Unique methods can detect events that might be missed by other pipelines, while joint detections provide stronger evidence that an event is a genuine CBC. Secondly, mitigating non-Gaussian transient noise artifacts (glitches) remains an ongoing challenge. Glitches can produce high-SNR triggers, and pipelines must avoid generating alerts based on these while still identifying true CBC signals [6]. As both the detection rate of CBCs and the frequency of instrumental glitches have increased over time [4, 5], addressing glitches without excluding true signals is becoming

---

\* 221840237@smail.nju.edu.cn

ing increasingly important. A detection method capable of identifying signals while minimizing the impact of glitches, and correctly interpreting signals contaminated by glitches, would be ideal. Given these challenges, deep learning-based detection methods are a logical avenue for exploration [22].

Deep learning has already proven useful in enhancing the accuracy and latency of various gravitational wave data analysis tasks (e.g., [22–32]). For BBH detection, deep learning has shown promise in achieving sensitivity comparable to matched filtering pipelines in real detector noise [30–32]. However, applying deep learning to lower-mass signals like BNS mergers introduces additional challenges. BNS signals are present in detector data for  $\mathcal{O}(100)$  s at current sensitivity, meaning their signal power is significantly more spread out compared to BBH mergers with equivalent SNR. Strain-based BNS detection methods [33–37] must either truncate the input window, losing signal power, or make approximations during pre-processing, which limits sensitivity. Similarly, spectrogram-based detection methods [38, 39] face analogous limitations. As a result, a deep learning approach for BNS detection that matches the sensitivity of matched filtering pipelines has yet to be demonstrated.

In this work, we investigate the use of a neural network (NN)-based search pipeline for detecting BNS and BBH mergers in the SNR time series generated by matched filtering. One advantage of detecting signals in the SNR time series is that the CBC signal power is more condensed compared to the strain, which is particularly beneficial for the longer-duration BNS mergers. Moreover, SNR time series are readily available as data products from matched filtering pipelines, making their online implementation relatively straightforward. This work is further motivated by [40] and [34], where we demonstrated that BBH detection using SNR time series achieved promising sensitivity results, especially for lower-mass BBH systems. **Much of the context in Sec. I is derived from that study and has been adapted here just as part of course homework.**

The structure of the remainder of this work is as follows. In Sec. II we cover how we implement matched filtering, select our detector noise, and how we generate our template bank and training datasets. In Sec. III we cover the high-level architecture of our neural network and its training and validation. The comparison we use to run our search pipeline by different models, mergers and methods and assign false alarm rates is presented in Sec. IV. In Sec. V we summarise the findings of this work, discuss their implications, and discuss potential future improvements.

## II. DATASET GENERATION

In this section, we introduce the concept of matched filtering and discuss how we use matched filtering to generate our training and validation datasets. In Sec. II A we

define our implementation of matched filtering. Sec. II B describes how we generate the BNS and BBH template bank used in the rest of this work. In Sec. II C we create our training and validation datasets for our neural network.

### A. Matched filtering

Matched filtering is a signal processing technique widely used in gravitational wave research, as it provides the optimal method for detecting modeled signals in stationary Gaussian noise [41]. This technique involves cross-correlating a signal template  $s$  with incoming detector data  $h$ , resulting in a signal-to-noise ratio (SNR) time series  $\rho(t)$  [41, 42]:

$$\rho^2(t) = \frac{z(t)}{\langle s|s \rangle}, \quad (1)$$

where  $\langle s|s \rangle$  represents the noise-weighted inner product of the template, and  $z(t)$  is the matched filter:

$$z(t) = 4 \int_{f_{\text{low}}}^{f_{\text{high}}} \frac{\tilde{s}(f)\tilde{h}^*(f)}{S_n(f)} e^{2\pi i f t} df, \quad (2)$$

Here,  $S_n(f)$  is the estimated one-sided power spectral density (PSD) of the detector noise, and a tilde denotes the Fourier transform of the template or data.

The SNR time series offers a significant advantage as input to a deep learning model compared to detector strain data. In gravitational wave strain data, a CBC signal can span hundreds of seconds, depending on the progenitor masses and the low-frequency sensitivity of the interferometer. Matched filtering condenses this signal power into an SNR peak that is only tens of milliseconds wide, simplifying detection.

We implemented matched filtering using Python’s NumPy module [43], enabling efficient batch computation of SNR time series through array-wise operations. Our implementation is adapted from the PyCBC library [44].

### B. Template bank generation

We generated our template bank using the LIGO Algorithm Library (LALSuite [45]). For BNS signals, we employed the IMRPhenomPv2\_NRTidalv2 waveform model [46], simulating systems with component masses ranging from 1 to  $2.6 M_\odot$ . These simulations include tidal deformation contributions characterized by the tidal deformability parameter,  $\Lambda$ . Instead of relying on a specific equation of state (EoS) for calculating  $\Lambda$ , we generalized the  $M - \Lambda$  relationship following the approach in [47]. To define a broad region in the  $M - \Lambda$  space, we used two fit functions,  $g_1(x) = a_1 \exp(b_1 x)$  and  $g_2(x) = a_2 \exp(b_2 x)$ , as upper and lower bounds to several microphysical EoS

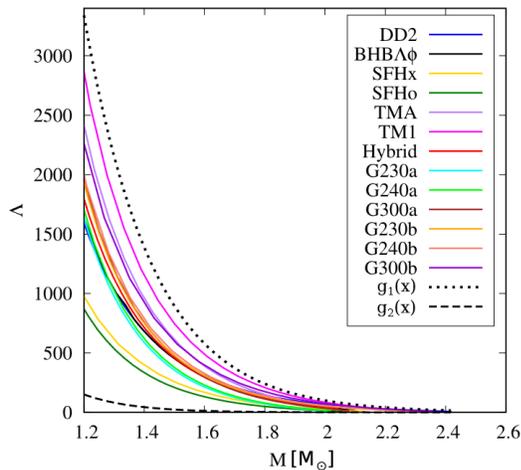


FIG. 1.  $M - \Lambda$  curves of several EoSs taken from [48] are shown as solid colored curves. The black dotted and dashed curves labeled as  $g_1(x)$  and  $g_2(x)$ , respectively, are chosen functions to span a wide range in the  $M - \Lambda$  space. The  $M - \Lambda$  data points enveloped in this region are used to simulate GW signals of BNS mergers in the regression task.

models, as illustrated in Figure 1. For each primary mass,  $m_1$ , the corresponding tidal deformability,  $\Lambda_1$ , was randomly sampled from the range defined by the envelope between the dotted ( $g_1(x)$ ) and dashed ( $g_2(x)$ ) lines in Figure 1. Specifically,  $\Lambda_1$  is assigned a value between  $g_1(m_1)$  and  $g_2(m_1)$ , using fit parameters  $a_1 = 6.45 \times 10^5$ ,  $b_1 = -4.386$ ,  $a_2 = 2.45 \times 10^5$ , and  $b_2 = -6.16$ . The secondary mass,  $m_2$ , was then assigned a tidal deformability,  $\Lambda_2$ , from a uniform distribution within the range  $(\Lambda_1, g_1(m_2)]$ . This ensures  $\Lambda_1$  monotonically decreases with mass in the region of interest. It is worth noting that not all EoS models shown in Figure 1 necessarily satisfy the  $\Lambda$  constraints derived from GW170817 [48]. However, this choice of envelope ensures that the  $M - \Lambda$  range of relevance is enclosed by the parameters used in our simulations.

For BBH signals, we used the IMRPhenomD waveform model [49], which describes the inspiral-merger-ringdown templates for systems with non-precessing spins. We simulated systems with component masses between 5 and 100  $M_\odot$  and assumed zero spin. The BBH signals were simulated with a duration of 1 second sampled at 8192 Hz, while the BNS signals were simulated with a duration of 10 seconds sampled at 4096 Hz. This distinction accounts for the longer duration and higher frequencies typically present in BNS signals and reduces memory requirements during neural network training.

The input parameters for generating the template bank are shown in Table I. We set the maximum z-aligned spin magnitude  $S_z$  to 0.5, as it is unlikely neutron stars would merge with a larger spin [50, 51].

Parameter	BNS	BBH
Minimum component mass ( $M_\odot$ )	1	5
Maximum component mass ( $M_\odot$ )	2.6	100
Maximum $ S_z $	0.5	0
Lower frequency cutoff (Hz)	30	12
Approximant	NRTidalv2	IMRPhenomD
Observation time (s)	1	10
Sampling rate (Hz)	4096	8192
SNR	30	20

TABLE I. Parameters used to create the template bank. All mass units are in  $M_\odot$ .

### C. Training dataset construction

Regretfully, we just use stimulated Gaussian noise instead of obtaining real LIGO data from the LIGO GW Open Science Center (GWOSC). Both the data and the simulated signals are whitened separately with power spectral density (PSD) computed directly from the raw GW strain data by Welch’s method. Whitening of data is an operation of rescaling the noise contribution at each frequency to have equal power. Because whitening is a linear procedure, whitening both parts individually is equivalent to whitening their sum. The waveforms are subsequently shifted such that the peak amplitude of each waveform is randomly positioned in the range from the 60% to 80% part of the time series, to reassure robustness of the network against temporal translations.

Example time series are shown in Fig. 2. The training sets for the detection consist of 10000 independent time series with 1/2 containing BNS or BBH signal + noise, and 1/2 noise only and the testing data sets consist of 1000 samples. We use two data generations for training, method 1 typically involves the traditional offline approach, where all augmented data is pre-generated and stored before training. However, this method is time-consuming and requires significant storage space to save the augmented dataset, making it unsuitable for large-scale datasets or real-time applications. In contrast, method 2: On-the-fly data augmentation dynamically generates augmented data in real time during training. This approach not only avoids the high computational and storage costs of pre-computation but also allows for randomized augmentation strategies in each training cycle, providing a diverse set of input samples and significantly improving the model’s generalization ability. We found that by starting neural network training at high SNR ( $\geq 50$ ) and then gradually increasing the noise in each subsequent training session until the final SNR is in the range between 3 and 20, we found that the performance of the detection CNN can be quickly maximized at low SNR (typically after only 10 epochs) while retaining performance at high SNR. That’s why we choose high SNR (SNR=20 and 30 for BBH and BNS respectively) to generate training data and test data for ensuring basic

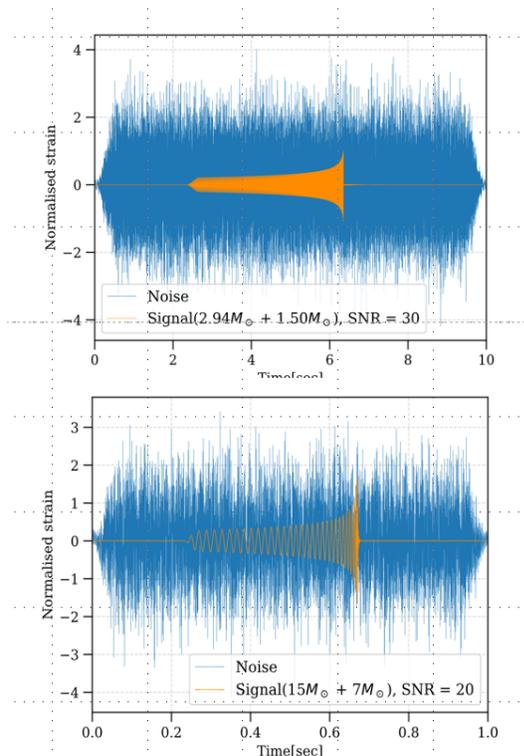


FIG. 2. Inserted into simulated LIGO noise are representative signals from binary neutron star (BNS) and binary black hole (BBH) systems. (Upper panel) Presented is a whitened, noise-free temporal sequence representing a gravitational-wave signal from a binary neutron star system with constituent masses of  $m_1 = 2.94M_\odot$  and  $m_2 = 1.5M_\odot$ , featuring an optimal signal-to-noise ratio (SNR) of 30. The blue curve illustrates the same gravitational-wave signal superimposed with additive whitened simulated LIGO noise with unit variance. This time series serves as an exemplar within the dataset utilized for training, validation, and testing of the convolutional neural network. (Lower panel) Analogous to the upper panel, this display pertains to a binary black hole gravitational-wave signal characterized by component masses of  $m_1 = 15M_\odot$  and  $m_2 = 7M_\odot$ , with an optimal SNR of 20. (Note:  $\Lambda = 0$  for black holes.)

performances of our models.

### III. THE NEURAL NETWORK

#### A. Model architecture

The convolutional neural networks (CNN) and Residual networks (ResNet50, ResNet101, and ResNet152) serve as the foundational models for the two-detector system discussed. ResNet addresses the vanishing or exploding gradient problem commonly encountered in deep networks by introducing residual learning. Instead of directly mapping the input to the desired output  $H(x)$ , ResNet learns the residual function  $F(x) = H(x) - x$ ,

enabling the network to model  $H(x) = F(x) + x$ . This formulation facilitates efficient training of very deep neural networks by preserving gradient flow through identity shortcut connections, which bypass one or more layers without additional parameters. These shortcuts mitigate the degradation problem, ensure stable gradient propagation, and allow deeper architectures to achieve improved representation learning and generalization performance. ResNet’s innovative design has set a new standard for deep network architectures in terms of scalability and accuracy. The overarching architecture consists of several independent branches of residual blocks, each corresponding to the SNR time series of a distinct interferometer. The primary distinction among the three ResNet models lies in the number of residual blocks they contain. As depicted in Figure III, a typical ResNet architecture comprises four residual layers, with varying quantities of residual blocks across each layer depending on the model. Specifically, in the ResNet architectures, the configurations are as follows: ResNet([3, 4, 6, 3]), ResNet([3, 4, 23, 3]), and ResNet([3, 8, 36, 3]). The architecture of the detection CNN is notably simpler. It is composed of three convolutional layers interspersed with three pooling layers, culminating in two fully connected dense layers. The respective filter sizes for the convolutional layers are 16, 32, 64, and 128, while the dense layers have sizes of 64 and 2, as illustrated in Figure II. The initial layer serves as the neural network’s input, which is a one-dimensional time-series vector (having a dimension of 8192 for BBH scenarios). The network concludes with a softmax output layer that calculates the inferred class probabilities.

In the course of training, we incorporate a dropout layer between each dense layer within the combiner sub-network to mitigate overfitting on the training dataset. Subsequently, a sigmoid activation layer is applied following the final dense layer to restrict predictions to a range between 0 and 1. Notably, omitting this activation during inference has demonstrated effectiveness in addressing the resolution constraints inherent to 32-bit precision [52], thereby allowing our ranking statistic to remain unbounded. Additionally, a custom layer is introduced before the sigmoid activation layer, which divides the output of the last dense layer by a factor of 4. This measure serves to prevent the sigmoid layer from rounding predictions to 0 or 1 during the training phase.

#### B. Training

To construct and train the neural networks, we employed PyTorch, which offers a high-level programming framework to interface with the PyTorch deep-learning library (<https://pytorch.org/>). The network’s training utilized the training and validation datasets detailed in Sec. II C. For optimization, we implemented stochastic gradient descent with an adaptive learning rate, utilizing the ADAM approach [53] alongside the AMSgrad modi-

Layer	Array Type	Size
Input	Matrix	$1 \times 2 \times 16384$
Conv2D + ELU + BatchNorm	Matrix	$8 \times 2 \times 16353$
MaxPool2D	Matrix	$8 \times 2 \times 2044$
Conv2D + ELU + BatchNorm	Matrix	$16 \times 2 \times 2029$
Conv2D + ELU + BatchNorm	Matrix	$16 \times 2 \times 2014$
Conv2D + ELU + BatchNorm	Matrix	$32 \times 2 \times 1999$
Conv2D + ELU + BatchNorm	Matrix	$64 \times 2 \times 1992$
MaxPool2D	Matrix	$64 \times 2 \times 332$
Conv2D + ELU + BatchNorm	Matrix	$64 \times 2 \times 325$
Conv2D + ELU + BatchNorm	Matrix	$128 \times 2 \times 322$
Conv2D + ELU + BatchNorm	Matrix	$128 \times 2 \times 319$
MaxPool2D	Matrix	$128 \times 2 \times 79$
Flatten	Vector	20224
Linear + ELU + Dropout	Vector	64
Linear	Vector	2
Output	Vector	2

TABLE II. The Architecture of CNN

Layer	Array Type	Channels
Input	Matrix	1
Conv2D + ReLU + BatchNorm	Matrix	64
MaxPool2D	Matrix	64
Residual Layer 1	Matrix	256
MaxPool2D	Matrix	256
Residual Layer 2	Matrix	512
MaxPool2D	Matrix	512
Residual Layer 3	Matrix	1024
MaxPool2D	Matrix	1024
Residual Layer 4	Matrix	2048
AvgPool2D	Matrix	2048
Flatten	Vector	-
Linear + ReLU + Dropout	Vector	-
Linear + ReLU + Dropout	Vector	-
Linear	Vector	-
Output	Vector	2

TABLE III. The Architecture of ResNet

fication. In the training of the neural networks, an initial learning rate of 0.001 was employed, with batch sizes designated as 16 for BBH and 8 for BNS. The training process was conducted using an NVIDIA GeForce RTX 4090, and the mini-batch size was dynamically adjusted based on the characteristics of the GPU and the datasets. Sparse categorical cross-entropy was utilized as the loss function.

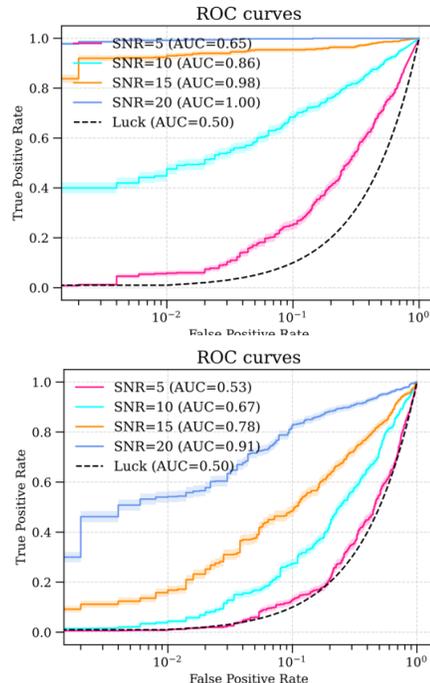


FIG. 3. ROC curves for test data sets containing BBH(upper) and BNS(lower) GW signals with optimal SNR,opt = 5, 10, 15, 20. The true alarm probability is shown versus the false alarm probability estimated from the output of the convolutional neural network.

#### IV. RESULTS

We assess the performance of the detection neural network by constructing and examining the receiver operator characteristic (ROC) curves for the BBH and BNS signal classes, for a given SNR. A ROC curve represents the fraction of signals identified correctly as their respective class, BBH or BNS (true alarm probability), versus the fraction of samples identified incorrectly as signals of the particular class (false alarm probability). We calculate the ROC curves with the Python scikit-learn library (<https://scikitlearn.org>), which constructs empirical ROC curves. An empirical ROC curve is a plot of the true alarm probability (TAP) versus the false alarm probability (FAP) for all possible thresholds, that is, each point on the ROC curve represents a different cut-off value. Thresholds that result in low FAP also tend to result in low TAP. As the TAP increases, the FAP increases as well. A ranking statistic is considered superior to another if at a fixed FAP it reaches a higher TAP (or sensitivity). We varied the optimal SNR from 1 to 20 in integer steps of 1 and the classifier was applied to inputs with approximately equal fractions of each GW signal class (Noise, BBH Signal, BNS Signal).

Fig.3 shows the ROC curves calculated for test data sets containing BBH and BNS GW signals. These re-

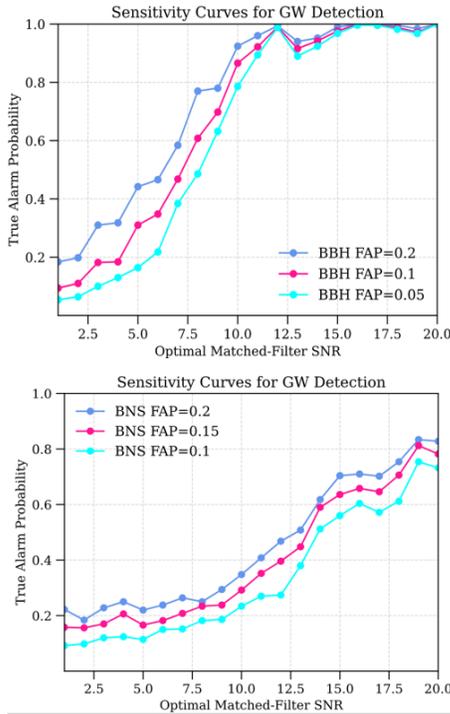


FIG. 4. Sensitivity curves illustrating the ability of the neural network to identify BNS(lower) and BBH(upper) GW signals. The true alarm probability is plotted as a function of the optimal SNR for false alarm probability. The figure shows the sensitivity of detecting GW signals embedded in Gaussian noise from the test data set.

sults are similar to the corresponding ROC curves in the case of simulated Gaussian noise [33], and indicate that the neural network is again more sensitive to detecting GWs from BBH than BNS mergers. It is seen that the CNN achieves a maximal sensitivity for BBH signals with optimal SNR 10 for  $FAP \geq 10^{-1}$  (Fig.4,upper window). On the other hand, it don't reaches a maximal sensitivity for BNS signals with optimal SNR=30 (Fig.4, lower window), namely our model's performance doesn't reach that level as in [33], which is mostly caused by the insufficiency of dataset quantity. Note that since the TAP is a function of the FAP, it also reaches a maximal sensitivity for BNS signals with lower optimal SNR at a higher FAP. For instance, at  $SNR_{opt} = 14$  the performance is similar to that at  $SNR_{opt} = 18$ .

Figure 5 illustrates the comparative analysis of methods applied in BBH using the CNN model. Traditional approaches(upper) exhibit considerable robustness concerning training accuracy and testing performance than on-the-fly data augmentation(lower). Notably, Method 1 exhibits slightly enhanced performance attributed to its extensive pre-computed dataset that offers improved feature space coverage. Nonetheless, this benefit results in a substantially higher computational time ( $T_1 \gg T_2$ ) and increased storage demands, as the method necessitates the pre-generation and storage of all augmented

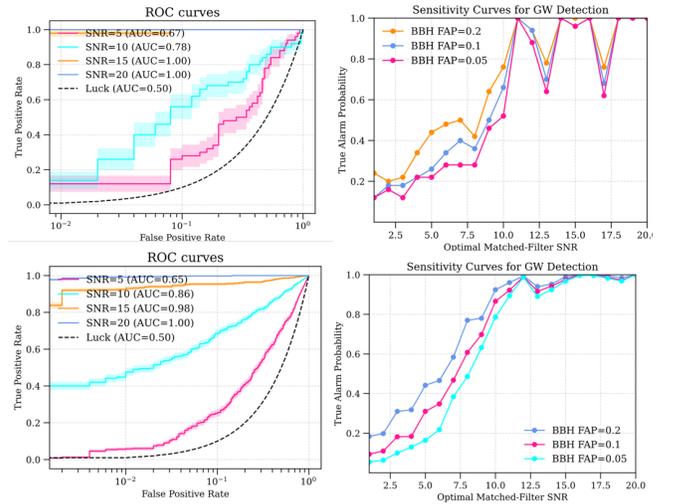


FIG. 5. Method comparison illustrates the comparative analysis of methods applied in BBH using the CNN model. Traditional approaches(upper) exhibit considerable robustness concerning training accuracy and testing performance than on-the-fly data augmentation(lower).

datasets. Conversely, Method 2 diminishes computational demands by dynamically producing augmented data throughout the training process, as demonstrated by the expedited convergence in training and its comparable efficacy under reduced SNR conditions. The ROC curves and sensitivity analyses indicate that although Method 1 attains marginally higher AUC values and sensitivity at constant false-alarm probabilities, Method 2 maintains competitiveness, especially at elevated SNR levels. The equilibrium between efficiency and performance renders on-the-fly data augmentation a compelling option for scenarios with restricted resources.

For the model comparison, The sensitivity curves for gravitational wave (GW) detection using CNN, ResNet50, and ResNet101 models actually demonstrate no too much differences in performance across varying false-alarm probabilities (FAP) and signal-to-noise ratios (SNR) as we see in Figure.6. The three models exhibits reasonable sensitivity which improves as the SNR increases. The additional depth of ResNet seems don't allows for more comprehensive feature representation, enabling consistent and robust performance even at lower SNRs. Whereas, Figure.6 is adopted the method 1(tradition data generation). When we compare these models in method 2, the results indicate that ResNet-based architectures significantly outperform CNNs in GW detection tasks, with ResNet101 providing optimal sensitivity at the cost of increased computational complexity compared to ResNet50. This highlights the importance of deeper architectures in achieving enhanced performance for challenging detection scenarios.

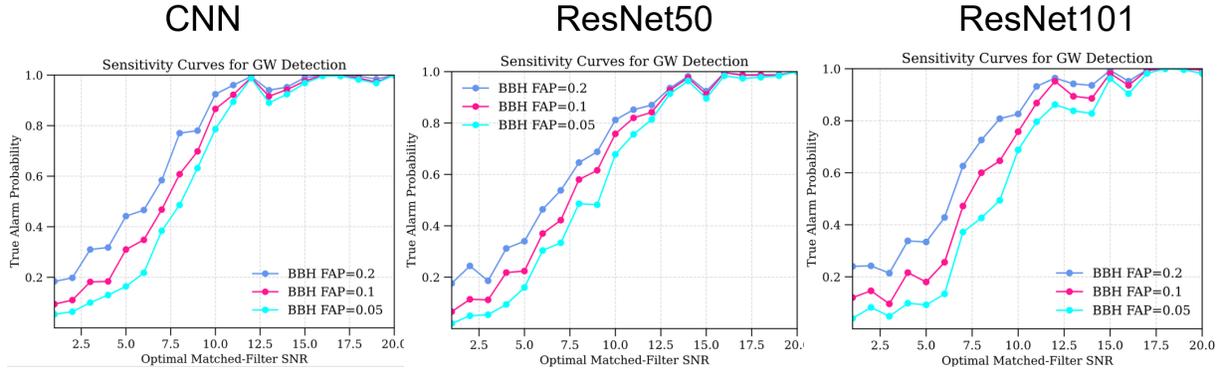


FIG. 6. Model comparison

## V. SUMMARY AND DISCUSSION

In this study, we employ deep learning techniques for gravitational wave (GW) detection tasks, illustrating their potential to attain high sensitivity and robustness in identifying signals from binary black hole (BBH) and binary neutron star (BNS) mergers. We assess the performance of neural networks using CNN, ResNet50, ResNet101, and ResNet152 architectures, comparing their sensitivity curves across various false alarm probabilities (FAPs) and signal-to-noise ratios (SNRs). Additionally, through the utilization of on-the-fly data augmentation and analysis via receiver operating characteristic (ROC) curves, this work offers insights into the balance between computational efficiency and detection efficacy.

The training dataset was carefully assembled to include 5000 samples of both signal-plus-noise and pure noise, maintaining consistent SNR values of 20 for BBH signals and 30 for BNS signals. This configuration facilitated the exposure of the network to a diverse array of samples, promoting generalization. Meanwhile, the testing dataset contained 1,000 samples and was specifically designed to evaluate the model's proficiency in accurately classifying previously unseen data. Although this configuration represents a substantial advancement towards effective GW detection, the relatively small dataset, particularly regarding BNS signals, seems to have restricted the models' efficacy. This limitation is most apparent in the less-than-optimal sensitivity observed for BNS detections, underscoring the necessity for additional data augmentation or alternative strategies to improve the representation of BNS signals in the training set.

The empirical ROC curves developed in this research demonstrate that the models attained superior TAPs for BBH signals relative to BNS signals at analogous FAPs. For instance, the CNN model achieves peak sensitivity for BBH signals at an optimal SNR of 10 when  $FAP \geq 0.1$ . Nonetheless, the model fails to attain a comparable maximum sensitivity for BNS signals, even when the SNR is optimally set at 30. This gap highlights the dif-

ficulty in detecting BNS signals, potentially due to their inadequate representation in the training dataset and the inherent disparities in waveform characteristics between BBH and BNS mergers.

It is also notable that at lower SNRs, the TAP for BNS signals remains similar for a range of optimal SNR values (e.g.,  $SNR=14$  and  $SNR=18$ ). This implies that the network's sensitivity does not significantly improve with increasing SNR under certain conditions, pointing to potential limitations in the network's feature extraction capabilities or the quality of the training data. These findings align with previous works, such as [33], which report comparable challenges in detecting BNS signals in simulated Gaussian noise.

Integrating on-the-fly data augmentation markedly enhanced training efficiency and diminished computational overhead relative to conventional pre-computed data generation techniques. Although Method 1 (traditional augmentation) yields slightly superior area under the curve (AUC) scores and sensitivity at fixed False Alarm Probabilities (FAPs) because of its comprehensive feature space coverage, it incurs considerable computational and storage requirements. In contrast, Method 2 (on-the-fly augmentation) dynamically generates augmented data during training, resulting in faster convergence and competitive performance, especially under reduced SNR conditions. The trade-off between efficiency and performance is critical for resource-constrained environments. The results indicate that Method 2 is a compelling alternative for scenarios where computational resources or storage capacity are limited. Additionally, its ability to maintain competitive sensitivity at higher SNR levels makes it a viable choice for real-time GW detection applications. These observations highlight the importance of selecting appropriate data augmentation techniques to balance computational requirements and model performance.

The sensitivity curves for CNN, ResNet50, ResNet 101 and ResNet152 architectures reveal notable differences in their ability to detect GW signals. While all three models demonstrate reasonable sensitivity improvements

with increasing SNR, the ResNet architectures consistently outperform the CNN, particularly at lower FAPs. ResNet50 and ResNet101 leverage their deeper architectures and residual connections to enhance feature representation and mitigate vanishing gradient issues, resulting in higher TAPs at fixed FAPs. However, the additional depth of ResNet101 does not always translate to significant performance gains compared to ResNet50, particularly when using traditional data augmentation methods (Method 1). This observation suggests that the marginal benefits of increased depth may diminish beyond a certain point, especially when computational complexity and training time are considered. On the other hand, when employing on-the-fly data augmentation (Method 2), ResNet101 achieves optimal sensitivity, highlighting the synergy between advanced architectures and efficient data augmentation strategies.

Despite the promising results, several challenges and limitations warrant further investigation:

- **Dataset Quantity and Diversity:** The limited size and diversity of the training dataset, particularly for BNS signals, constrain the models' ability to generalize. Future studies should focus on generating larger and more diverse datasets, potentially incorporating real LIGO data to enhance the realism and variability of training samples.
- **BNS Signal Detection:** The suboptimal performance of the models in detecting BNS signals raises questions about their ability to capture the unique features of these waveforms. Investigating alternative network architectures or feature extraction techniques tailored to BNS signals could improve sensitivity.
- **SNR Effects:** The sensitivity analyses indicate that the models' performance varies significantly with SNR. Further studies are needed to explore the relationship between SNR and detection accuracy, including adjustments to network hyperparameters and training strategies to optimize performance across a broader SNR range.
- **Model Optimization:** While ResNet101 achieves

the highest sensitivity, its increased computational complexity poses challenges for real-time applications. Exploring lightweight architectures or pruning techniques could help achieve a balance between performance and efficiency.

- **Combined Detection of BBH and BNS Signals:** Integrating BBH and BNS detection into a unified framework could streamline GW detection workflows and enhance overall efficiency. Developing models capable of distinguishing between these classes while maintaining high sensitivity is an important avenue for future research.

This study demonstrates the potential of deep learning approaches for GW detection, leveraging CNN and ResNet architectures to identify signals from BBH and BNS mergers. The analysis of sensitivity curves and ROC metrics reveals that while ResNet-based models outperform CNNs, their performance for BNS signals remains suboptimal, primarily due to dataset limitations. On-the-fly data augmentation emerges as a promising technique to enhance training efficiency and maintain competitive performance, particularly under reduced SNR conditions.

The findings underscore the importance of dataset quality, network architecture, and data augmentation methods in achieving robust GW detection. Future work should focus on addressing the challenges of dataset diversity, BNS signal detection, and model optimization, with a particular emphasis on integrating real LIGO data and exploring unified frameworks for detecting BBH and BNS signals. These efforts will contribute to the development of scalable and efficient deep learning-based GW detection systems, paving the way for advancements in astrophysical research and multi-messenger astronomy.

## VI. DATA AND CODES AVAILABILITY

The code we developed is based on <https://github.com/iphysresearch/GWData-Bootcamp>. This paper also motivated and referred by [https://iphysresearch.github.io/PhDthesis\\_html/](https://iphysresearch.github.io/PhDthesis_html/).

- 
- [1] LIGO Scientific Collaboration *et al.*, Advanced LIGO, *Classical and Quantum Gravity* **32**, 074001 (2015), [arXiv:1411.4547](https://arxiv.org/abs/1411.4547) [gr-qc].
- [2] F. Acernese *et al.*, Advanced Virgo: a second-generation interferometric gravitational wave detector, *Classical and Quantum Gravity* **32**, 024001 (2015), [arXiv:1408.3978](https://arxiv.org/abs/1408.3978) [gr-qc].
- [3] B. P. Abbott *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs, *Phys. Rev. X* **9**, 031040 (2019).
- [4] R. Abbott *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), GWTC-2: Compact Binary Coalescences Observed by LIGO and Virgo During the First Half of the Third Observing Run, *Phys. Rev. X* **11**, 021053 (2021), [arXiv:2010.14527](https://arxiv.org/abs/2010.14527) [gr-qc].
- [5] B. P. Abbott *et al.* (LIGO Scientific Collaboration, Virgo Collaboration, and KAGRA Collaboration), GWTC-3: Compact Binary Coalescences Observed by LIGO and Virgo during the Second Part of the Third Observing Run, *Physical Review X* **13**, 041039 (2023), [arXiv:2111.03606](https://arxiv.org/abs/2111.03606) [gr-qc].

- [6] B. Abbott *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral, *Phys. Rev. Lett.* **119**, 161101 (2017), [arXiv:1710.05832 \[gr-qc\]](#).
- [7] B. P. Abbott *et al.*, GW190425: Observation of a Compact Binary Coalescence with Total Mass  $\sim 3.4 M_{\odot}$ , *Astrophys. J. Lett.* **892**, L3 (2020), [arXiv:2001.01761 \[astro-ph.HE\]](#).
- [8] LIGO Scientific Collaboration *et al.*, Multi-messenger Observations of a Binary Neutron Star Merger, *Astrophys. J. Lett.* **848**, L12 (2017), [arXiv:1710.05833 \[astro-ph.HE\]](#).
- [9] A. Goldstein *et al.*, An Ordinary Short Gamma-Ray Burst with Extraordinary Implications: Fermi-GBM Detection of GRB 170817A, *Astrophys. J., Lett.* **848**, L14 (2017), [arXiv:1710.05446 \[astro-ph.HE\]](#).
- [10] D. Haggard, M. Nynka, J. J. Ruan, V. Kalogera, S. B. Cenko, P. Evans, and J. A. Kennea, A Deep Chandra X-Ray Study of Neutron Star Coalescence GW170817, *ApJ* **848**, L25 (2017), [arXiv:1710.05852 \[astro-ph.HE\]](#).
- [11] B. P. Abbott *et al.*, A gravitational-wave standard siren measurement of the Hubble constant, *Nature* **551**, 85 (2017), [arXiv:1710.05835 \[astro-ph.CO\]](#).
- [12] D. Radice, A. Perego, F. Zappa, and S. Bernuzzi, GW170817: Joint constraint on the neutron star equation of state from multimessenger observations, *The Astrophysical Journal* **852**, L29 (2018).
- [13] L. Baiotti, Gravitational waves from neutron star mergers and their relation to the nuclear equation of state, *Progress in Particle and Nuclear Physics* **109**, 103714 (2019).
- [14] T. Akutsu *et al.*, Overview of KAGRA: Detector design and construction history, *Progress of Theoretical and Experimental Physics* **2021**, 05A101 (2020), <https://academic.oup.com/ptep/article-pdf/2021/5/05A101/37974994/ptaa125.pdf>.
- [15] C. Messick *et al.*, Analysis framework for the prompt discovery of compact binary mergers in gravitational-wave data, *Physical Review D* **95**, 042001 (2017), [arXiv:1604.04324 \[astro-ph.IM\]](#).
- [16] A. H. Nitz, T. Dent, T. Dal Canton, S. Fairhurst, and D. A. Brown, Detecting Binary Compact-object Mergers with Gravitational Waves: Understanding and Improving the Sensitivity of the PyCBC Search, *Astrophys. J.* **849**, 118 (2017), [arXiv:1705.01513 \[gr-qc\]](#).
- [17] T. Adams, D. Buskulic, V. Germain, G. M. Guidi, F. Marion, M. Montani, B. Mours, F. Piergiovanni, and G. Wang, Low-latency analysis pipeline for compact binary coalescences in the advanced gravitational wave detector era, *Classical and Quantum Gravity* **33**, 175012 (2016), [arXiv:1512.02864 \[gr-qc\]](#).
- [18] Q. Chu, M. Kovalam, L. Wen, T. Slaven-Blair, J. Bosveld, Y. Chen, P. Clearwater, A. Codoreanu, Z. Du, X. Guo, X. Guo, K. Kim, T. G. F. Li, V. Oloworaran, F. Panther, J. Powell, A. S. Sengupta, K. Wette, and X. Zhu, SPIIR online coherent pipeline to search for gravitational waves from compact binary coalescences, *Phys. Rev. D* **105**, 024023 (2022), [arXiv:2109.14183 \[gr-qc\]](#).
- [19] S. Klimentenko *et al.*, Method for detection and reconstruction of gravitational wave transients with networks of advanced detectors, *Physical Review D* **93**, 042004 (2016), [arXiv:1511.05999 \[gr-qc\]](#).
- [20] B. Allen,  $\chi^2$  time-frequency discriminator for gravitational wave detection, *Phys. Rev. D* **71**, 062001 (2005).
- [21] T. Dal Canton, A. H. Nitz, B. Gadre, G. S. Cabourn Davies, V. Villa-Ortega, T. Dent, I. Harry, and L. Xiao, Real-time Search for Compact Binary Mergers in Advanced LIGO and Virgo's Third Observing Run Using PyCBC Live, *ApJ* **923**, 254 (2021), [arXiv:2008.07494 \[astro-ph.HE\]](#).
- [22] D. Beveridge, A. McLeod, L. Wen, and A. Wicenc, A Novel Deep Learning Approach to Detecting Binary Black Hole Mergers, *arXiv e-prints*, [arXiv:2308.08429 \(2024\)](#), [arXiv:2308.08429 \[gr-qc\]](#).
- [23] E. Cuoco *et al.*, Enhancing gravitational-wave science with machine learning, *Machine Learning: Science and Technology* **2**, 011002 (2021), [arXiv:2005.03745 \[astro-ph.HE\]](#).
- [24] M. Dax, S. R. Green, J. Gair, J. H. Macke, A. Buonanno, and B. Schölkopf, Real-Time Gravitational Wave Science with Neural Posterior Estimation, *Phys. Rev. Lett.* **127**, 241103 (2021), [arXiv:2106.12594 \[gr-qc\]](#).
- [25] H. Gabbard, C. Messenger, I. S. Heng, F. Tonolini, and R. Murray-Smith, Bayesian parameter estimation using conditional variational autoencoders for gravitational-wave astronomy, *Nature Physics* **18**, 112 (2022), [arXiv:1909.06296 \[astro-ph.IM\]](#).
- [26] C. Chatterjee, M. Kovalam, L. Wen, D. Beveridge, F. Diakogiannis, and K. Vinsen, Rapid Localization of Gravitational Wave Sources from Compact Binary Coalescences Using Deep Learning, *ApJ* **959**, 42 (2023), [arXiv:2207.14522 \[gr-qc\]](#).
- [27] S. Bahaadini, V. Noroozi, N. Rohani, S. Coughlin, M. Zevin, J. Smith, V. Kalogera, and A. Katsaggelos, Machine learning for gravity spy: Glitch classification and dataset, *Information Sciences* **444**, 172 (2018).
- [28] R. Essick, P. Godwin, C. Hanna, L. Blackburn, and E. Katsavounidis, iDQ: Statistical inference of non-gaussian noise with auxiliary degrees of freedom in gravitational-wave detectors, *Machine Learning: Science and Technology* **2**, 015004 (2020).
- [29] V. Skliris, M. R. K. Norman, and P. J. Sutton, Real-Time Detection of Unmodelled Gravitational-Wave Transients Using Convolutional Neural Networks, *arXiv e-prints*, [arXiv:2009.14611 \(2020\)](#), [arXiv:2009.14611 \[astro-ph.IM\]](#).
- [30] M. B. Schäfer, O. Zelenka, A. H. Nitz, H. Wang, S. Wu, Z.-K. Guo, Z. Cao, Z. Ren, P. Nousi, N. Stergioulas, P. Iosif, A. E. Koloniari, A. Tefas, N. Passalis, F. Salemi, G. Vedovato, S. Klimentenko, T. Mishra, B. Brüggemann, E. Cuoco, E. A. Huerta, C. Messenger, and F. Ohme, First machine learning gravitational-wave search mock data challenge, *Phys. Rev. D* **107**, 023021 (2023), [arXiv:2209.11146 \[astro-ph.IM\]](#).
- [31] P. Nousi, A. E. Koloniari, N. Passalis, P. Iosif, N. Stergioulas, and A. Tefas, Deep residual networks for gravitational wave detection, *Phys. Rev. D* **108**, 024022 (2023), [arXiv:2211.01520 \[gr-qc\]](#).
- [32] E. Marx, W. Benoit, A. Gunny, R. Omer, D. Chatterjee, R. C. Venterea, L. Wills, M. Saleem, E. Moreno, R. Raikman, E. Govorkova, D. Rankin, M. W. Coughlin, P. Harris, and E. Katsavounidis, A machine-learning pipeline for real-time detection of gravitational waves from compact binary coalescences, *arXiv e-prints* [10.48550/arXiv.2403.18661 \(2024\)](#), [arXiv:2403.18661 \[gr-qc\]](#).

- [33] P. G. Krastev, Real-time detection of gravitational waves from binary neutron stars using artificial neural networks, *Physics Letters B* **803**, 135330 (2020).
- [34] P. G. Krastev, K. Gill, V. A. Villar, and E. Berger, Detection and parameter estimation of gravitational waves from binary neutron-star mergers in real LIGO data using deep learning, *Physics Letters B* **815**, 136161 (2021), [arXiv:2012.13101 \[astro-ph.IM\]](#).
- [35] M. B. Schäfer, F. Ohme, and A. H. Nitz, Detection of gravitational-wave signals from binary neutron star mergers using machine learning, *Physical Review D* **102**, 063015 (2020), [arXiv:2006.01509 \[astro-ph.HE\]](#).
- [36] G. Baltus, J. Janquart, M. Lopez, A. Reza, S. Caudill, and J.-R. Cudell, Convolutional neural networks for the detection of the early inspiral of a gravitational-wave signal, *Phys. Rev. D* **103**, 102003 (2021), [arXiv:2104.00594 \[gr-qc\]](#).
- [37] R. Qiu, P. G. Krastev, K. Gill, and E. Berger, Deep learning detection and classification of gravitational waves from neutron star-black hole mergers, *Physics Letters B* **840**, 137850 (2023), [arXiv:2210.15888 \[astro-ph.IM\]](#).
- [38] W. Wei and E. A. Huerta, Deep learning for gravitational wave forecasting of neutron star mergers, *Physics Letters B* **816**, 136185 (2021), [arXiv:2010.09751 \[gr-qc\]](#).
- [39] J. Aveiro, F. F. Freitas, M. Ferreira, A. Onofre, C. Providência, G. Gonçalves, and J. A. Font, Identification of binary neutron star mergers in gravitational-wave data using object-detection machine learning models, *Phys. Rev. D* **106**, 084059 (2022), [arXiv:2207.00591 \[astro-ph.IM\]](#).
- [40] A. McLeod, D. Beveridge, L. Wen, and A. Wicenc, A binary neutron star merger search pipeline powered by deep learning [10.48550/arXiv.2409.06266](#) (2024), [arXiv:2409.06266 \[gr-qc\]](#).
- [41] B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. E. Creighton, FINDCHIRP: An algorithm for detection of gravitational waves from inspiraling compact binaries, *Phys. Rev. D* **85**, 122006 (2012), [arXiv:gr-qc/0509116 \[gr-qc\]](#).
- [42] A. H. Nitz, T. Dal Canton, D. Davis, and S. Reyes, Rapid detection of gravitational waves from compact binary mergers with PyCBC Live, *Phys. Rev. D* **98**, 024050 (2018).
- [43] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, Array programming with NumPy, *Nature* **585**, 357 (2020).
- [44] A. Nitz, I. Harry, D. Brown, C. M. Biwer, J. Willis, T. D. Canton, C. Capano, T. Dent, L. Pekowsky, G. S. C. Davies, S. De, M. Cabero, S. Wu, A. R. Williamson, D. Macleod, B. Machenschalk, F. Pannarale, P. Kumar, S. Reyes, dfinstad, S. Kumar, M. Tápai, L. Singer, P. Kumar, B. U. V. Gadre, maxtrevor, veronica villa, S. Khan, S. Fairhurst, and K. Chandra, *gwastro/pycbc: v2.3.2 release of pycbc* (2023).
- [45] LIGO Scientific Collaboration, Virgo Collaboration, and KAGRA Collaboration, *LVK Algorithm Library - LAL-Suite*, Free software (GPL) (2018).
- [46] T. Dietrich, A. Samajdar, S. Khan, N. K. Johnson-McDaniel, R. Dudi, and W. Tichy, Improving the nrtidal model for binary neutron star systems, *Physical Review D* **100**, 10.1103/physrevd.100.044003 (2019).
- [47] S. Soma, H. Stöcker, and K. Zhou, Mass and tidal parameter extraction from gravitational waves of binary neutron stars mergers using deep learning, *Journal of Cosmology and Astroparticle Physics* **2024** (01), 009.
- [48] S. Soma and D. Bandyopadhyay, Properties of binary components and remnant in gw170817 using equations of state in finite temperature field theory models, *The Astrophysical Journal* **890**, 139 (2020).
- [49] S. Husa, S. Khan, M. Hannam, M. Pürrer, F. Ohme, X. J. Forteza, and A. Bohé, Frequency-domain gravitational waves from nonprecessing black-hole binaries. i. new numerical waveforms and anatomy of the signal, *Physical Review D* **93**, 10.1103/physrevd.93.044006 (2016).
- [50] D. A. Brown, I. Harry, A. Lundgren, and A. H. Nitz, Detecting binary neutron star systems with spin in advanced gravitational-wave detectors, *Phys. Rev. D* **86**, 084017 (2012), [arXiv:1207.6406 \[gr-qc\]](#).
- [51] X. Zhu, E. Thrane, S. Osłowski, Y. Levin, and P. D. Lasky, Inferring the population properties of binary neutron stars with gravitational-wave measurements of spin, *Phys. Rev. D* **98**, 043002 (2018).
- [52] M. B. Schäfer, O. Zelenka, A. H. Nitz, F. Ohme, and B. Brügmann, Training strategies for deep learning gravitational-wave searches, *Phys. Rev. D* **105**, 043002 (2022), [arXiv:2106.03741 \[astro-ph.IM\]](#).
- [53] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization (2017), [arXiv:1412.6980 \[cs.LG\]](#).