PROCEEDINGS OF SCIENCE

Auto-encoder model for faster generation of gravitational waveform approximations

Suyog Garg[a,b,*]

[a] *Department of Physics, The University of Tokyo*
Bunkyo-ku, Tokyo 113-0033 JAPAN

[b] *Research Center for Early Universe, The University of Tokyo,*
Bunkyo-ku, Tokyo 113-0033 JAPAN

E-mail: gargsuyog@g.ecc.u-tokyo.ac.jp

The LIGO-Virgo-KAGRA international collaboration have observed more than 250 gravitational wave events using their global network of observatories. Estimation of source parameters for each of these events requires about a million likelihood computations to properly constrain the posteriors. Each such computation entails solving the general relativity equations to obtain a theoretical waveform, which is then matched against the detected signal. This operation is computationally heavy, especially in the case of complex waveforms. The upcoming gravitational wave observatories, with an estimated $10^4 - 10^6$ detections per year, make it imperative to have solutions for the evident bottleneck for rapid parameter estimation. Towards this end, we present an auto-encoder model for generation of effective one-body SEOBNRv4 binary black hole waveforms. We train our model with $\sim 27,300$ samples. Our parameter space is made of the two binary component masses: $m_1, m_2 \in [5, 75]\ M_\odot$ with a hard mass ratio limit of $q = m_1/m_2 < 10$. Our model is able to generate 10^4 samples in O(1) second, with a median polarization mismatch value of order 10^{-3}. Our work provides the first step towards having a production ready framework for real-time rapid generation of highly-accurate gravitational waveform approximations. This will enable orders-of-magnitude faster online parameter estimation, while basically providing the same scientific potential.

39th International Cosmic Ray Conference (ICRC2025)
15–24 July 2025
Geneva, Switzerland

ICRC 2025
The Astroparticle Physics Conference
Geneva July 15-24, 2025

[*] Speaker

1. Introduction

Since the first detection of gravitational waves (GW) in 2015, the LIGO-Virgo-KAGRA collaboration have now detected signals from more than 250 compact binary sources. Each detection event observation comprises signal detection in noisy data and then estimation of the source parameters. For a single event, $\sim 10^6$ likelihood calculations of the detected signal and theoretical waveforms with known source parameters, can be required to accurately estimate the source posteriors. Various theoretical models for the waveforms exist, like, phenomenological, effective one-body, numerical relativity surrogates. In the upcoming years, the third generation GW observatories like the Einstein telescope and Cosmic Explorer, are expected to observe $10^4 - 10^6$ events per year. This leads to a bottleneck in the parameter estimation process, of likelihood computations of order 10^{10} which is computationally infeasible. Machine learning methods can use highly useful in solving this bottleneck [1].

In this paper, we demonstrate a machine learning model, based on the auto-encoder architecture, for generating faster and accurate effective one-body time-domain gravitational waveform approximations. The layout of the paper is as follows. Section 2 outlines the data processing steps, while Section 3 describes our model architecture and training procedures. Then in Section 4 we list our results. Finally in Section 5 we conclude by discussing future directions and further implications of our model.

2. Data Processing

Our parameter space consists of the two binary component masses $m_1, m_2 \in [5, 75]\, M_\odot$, with a hard mass ratio limit of $q = m_1/m_2 < 10$. We sample a total of 39,070 data-points from this parameter space, with a grid spacing of $\delta m = 0.25\, M_\odot$. These samples are then split into three mutually exclusive datasets for training, validation and testing with a 70%, 10% and 20% data fraction respectively. Our training (validation) consists of 27,349 (3907) samples. For each of set of parameters, we calculate the full inspiral-merger-ringdown (IMR) waveform using the effective one-body approximant SEOBNRv4 from `pyCBC`. SEOBNRv4 belongs to a family of aligned-spin numerical relativity waveforms for binary black holes in spin-precessing systems. These are our input training data and waveform reconstruction targets. Each of our data sample has 1 sec long duration with a sampling rate of 8196 Hz. The waveforms we currently focus on are for zero-spin systems and thus, only depend on the two mass parameters.

Now, the most important step in modelling any machine learning problem is proper data preparation and preprocessing. This becomes slightly tricky when our aim is that of training a model to generate gravitational waveform approximations. It turns out that for most Machine Learning model architectures, a full IMR time series is a relatively tougher problem to learn. The issue is exacerbated if we want to accurately extract features inherent in the original signal from the reconstructions. Therefore, we convert the IMR $h_{+,\times}(t)$ polarization time series to amplitude and instantaneous frequency. These are much simpler monotonous functions of time, which makes the learning the target easier for the model. From polarizations $h_{+,\times}(t)$, the amplitude $A(t)$ and instantaneous frequency $f(t)$ can be obtained via the instantaneous phase $\phi(t)$ as:

$$A(t) = \sqrt{h_+^2(t) + h_\times^2(t)}, \qquad \phi(t) = \arctan\left(h_+(t),\, h_\times(t)\right), \qquad f(t) = \frac{\phi(t+\Delta t) - \phi(t)}{2\pi\, \Delta t} \tag{1}$$

Note that by definition this means that the frequency series will have one sampled data point less than what the original polarization time series contained. See Figure 1 for our data composition.

We also normalize the amplitude and frequency series, so that the supplied input time series data to the model is within the same range of amplitudes and frequencies for all the data samples. This further simplifies the problem that the model has to learn.

$$\widehat{A}(t) = \frac{A(t) - \mu_A}{\sigma_A + \varepsilon}, \quad \widehat{f}(t) = \frac{f(t) - \mu_f}{\sigma_f + \varepsilon} \tag{2}$$

where, $\widehat{A}$ and $\widehat{f}$ are the normalized quantities, (μ_A, σ_A) and (μ_f, σ_f) are the normalization factors and ε is a small random number to keep values bounded near zero.

During the evaluation stage after our model has learned, when we are actually generating the output waveforms, we will denormalize the reconstructed amplitude and frequency series, calculate the phase and then reconstruct the polarizations, as follows:

$$h_+(t) = A(t)\cos\phi(t), \quad h_\times(t) = A(t)\sin\phi(t) \tag{3}$$

Thus, we will also pass along the values of the normalization factors and keep track of them alongside the input set of parameters for each data sample.

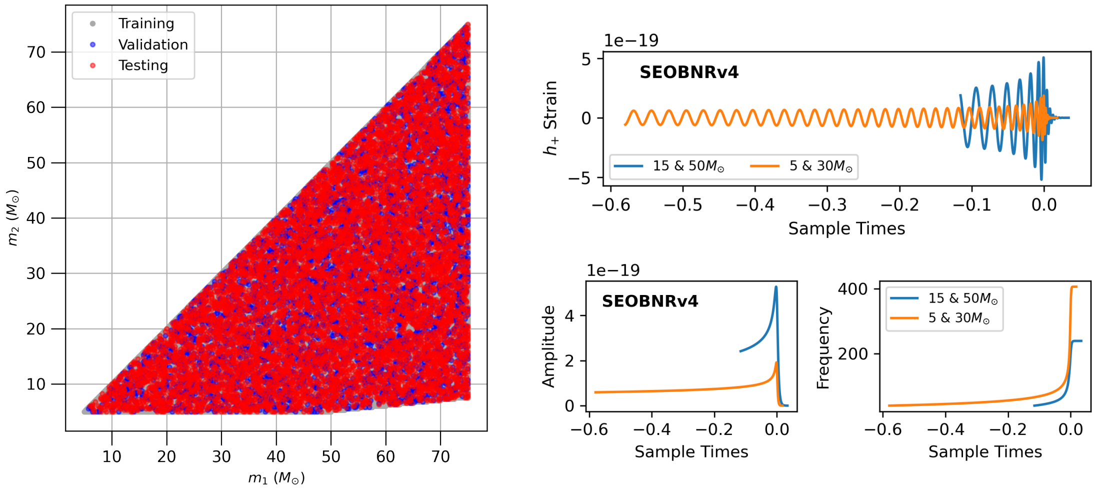

Figure 1: Left: Composition of our parameter space. Right: Representative sample of a gravitational waveform time series used for training and generation target. A gravitational polarization waveform can be converted to a amplitude and frequency series.

We also need to make sure that our model which takes only fixed length inputs actually has fixed length inputs supplied to it. The issue here is that sources with different binary component masses lead to waveforms of different durations. In principal, higher mass ratio binaries lead to shorter duration waveforms because the binary objects merge together faster. However, our neural network model can only take fixed length inputs. A typical idea to correct this issue is to append or prepend zeros at either end of a waveform to make them of the same length for all data samples. However, when we tried doing this for our data samples, since we were converting the polarization waveforms into the amplitude and frequency series, we found that the frequency series started to have some step-function like artifacts. This led to worse mismatch values later on during waveform reconstruction. Therefore, instead of prepending zeros to the input waveforms we come up with variable value for the lower frequency cut-off for different data samples. This ensures that even for higher mass binaries we have sufficiently long polarization waveforms.

The time to coalescence of a binary black hole system under the quadrupole approximation is given by:

$$t_c \propto f_{\text{low}}^{-8/3} \mathcal{M}^{-5/3} \tag{4}$$

where, f_{low} is the lower frequency cut-off and $\mathcal{M} = (m_1 m_2)^{3/5}/(m_1 + m_2)^{1/5}$ is the chirp mass. We estimate the proportionality constant k with by fixing f_{low} at 40 Hz for 500 samples with random chirp masses and calculating the time to coalescence. We then use this value of k to calculate the required lower frequency cut-off for a new waveform with an arbitrary chirp mass such that the waveform has our desired signal duration. To account for the ringdown part of the full IMR waveform, we further lower the obtained f_{low} by 20%. If the waveform goes beyond our desired length of duration, we truncate the initial phase of the early inspiral.

3. Network Architecture and Methodologies

Previous authors have used many different kinds of machine learning algorithms for applications to the gravitational wave sciences [1]. We find that given our task of teaching a model to learn and generate the gravitational waveform time series given some input source parameters, an auto-encoder (AE) model architecture is the best suited to the purpose. An AE model consists of an encoder and a decoder part, each of which can be an independent combination of a convolutional neural networks and fully-connected linear layers. The model takes in input data samples, encodes them into a latent space representation and then tries to decode back this representation to be as close a semblance of the input data as possible. Now, since our parameters span a preset prior range in the parameter space, the latent space representation for our model should also sample values from a distribution. We also need to assume a Gaussian distribution for the latent space representation, so that during the application stage our model is able to extrapolate beyond the set of parameters used for training. Such a model is now called a variational auto-encoder (VAE) network. Note that within a parameter space, our problem is deterministic. So, the simpler AE model is also expected to work equally well.

Further, we want the model generated waveforms to correspond to one particular set of parameters, instead of waveforms being generated with random source characteristics. Therefore, we need to pass along the values of the parameters corresponding to some input waveform alongside the input data sample. This conditions the output of the model to these input labels, so that the model knows that the latent representation for some input sample has some corresponding source parameters. This kind of an auto-encoder model then becomes the conditional variational auto-encoder (CVAE). The specific CVAE architecture of our model consists of 2 conditionals, 2 encoders and 1 decoder (2C2E1D). We normalize each input data sample and pass the normalizing factors (mean and standard deviation) to the second encoder as keys. These keys are conditioned on their own label priors and help in denormalizing the reconstructed outputs during inference stage.

The internal configuration of each of these modules is same as the 2C2E1D model in Liao and Lin [2], with one difference that both of our label conditionals have 4 fully-connected layers of size

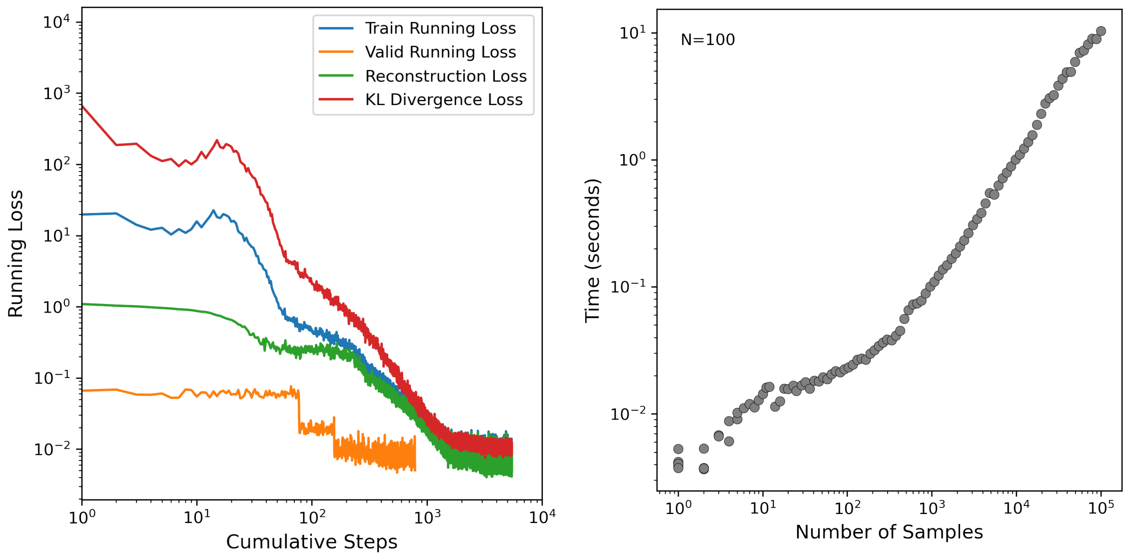

Figure 2: Left: Various loss function component values plotted against cumulative steps during the training and validation stages. Right: Time taken to generate x number of samples using our trained model during the evaluation or test stage. This is obtained by performing 100 random iterations with $x \in [1, 10^5]$.

500 each. The `ReLU` activation function is applied to all neural network layers. Our loss function or the objective to be minimized during the training process can be written as: $\mathcal{J} = \mathcal{L}_{recon} + \beta\mathcal{L}_{KL}$. Here the first term is the reconstruction loss, given by the mean squared error between the original and reconstructed waveforms; and the second term is the Kullback-Leibler divergence between the different latent space distributions learned by the encoders. We choose $\beta = 0.1$ as the KL loss weighting factor.

We train our model for 10 epochs with 547 data batches with 50 samples each from the training dataset per epoch. Validation is performed after each training epoch. We use Nvidia A100 80GB GPU for the training, with code written on `PyTorch` and run on a `cuda` environment. Our optimizer function is `ADAM` with initial learning rate of 10^{-4} and a learning rate update schedule with $\gamma = 0.1$ every 3 epochs, where γ is the rate of decay. Figure 2 (left) shows the trend of loss function optimization during the training and validation process.

4. Results and Discussion

Once our model has been trained, we test the model performance on the test dataset consisting of 7814 test samples. This dataset is chosen such that there are no repetitions from the training dataset, which ensures that the model is not overfitting within a closed parameter space. During the evaluation stage, we remove the encoders from the model, and only need the conditionals that take in a set of input parameters, pass it along to the decoder which reconstructs our desired amplitude and frequency series. From these reconstructed amplitude and frequencies, we obtain the polarization time series using Equation (3). Figure 2 (right) illustrates the speed of waveform generation for our trained model for 100 random iterations of number of samples generated between 1 and 10^5. It can be easily seen that our model is able to generate $\sim 10^4$ waveforms in O(1) second.

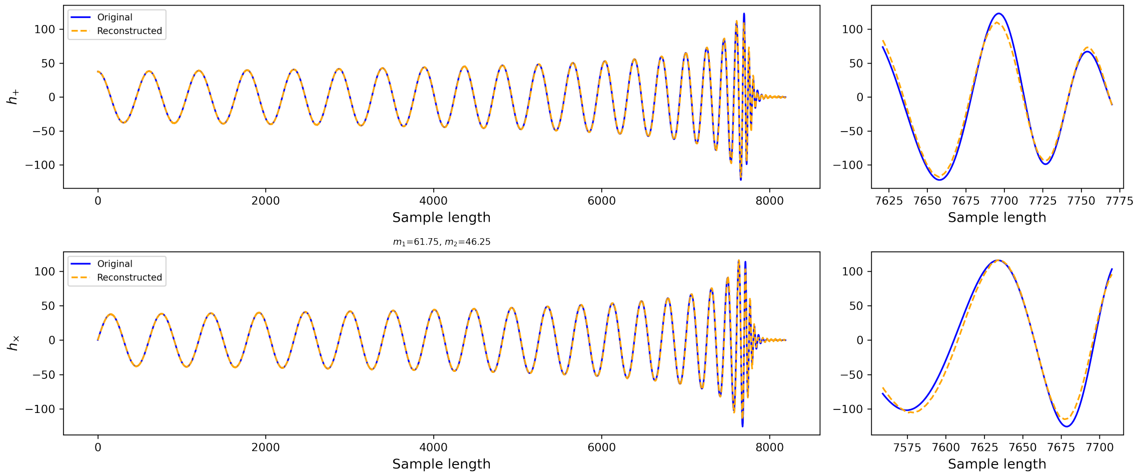

Figure 3: Overplot of the reconstructed waveform approximation with the original polarization time series for a randomly selected set of input parameters from the test dataset.

A qualitative overplot of the original and reconstructed polarization time series is shown in Figure 3. To evaluate how accurate the reconstructions generated from our model are, we then calculate the overlap and mismatch of the reconstructed waveforms with the original ones. For the amplitude and frequency series, which are real-valued vectors, we can use the cosine similarity (overlap) and cosine distance (mismatch) metrics as an evaluation criterion. For any between two discrete vectors a and b, these are respectively given by:

$$\mathcal{O}(a,b) = \frac{\langle a|b\rangle}{\sqrt{\langle a|a\rangle\langle b|b\rangle}}, \qquad \mathtt{MM}(a,b) = 1 - \mathcal{O}(a,b) \tag{5}$$

where, $\langle a|b\rangle = \sum_i a_i b_i$, is the inner product of the vectors and MM is the cosine distance or mismatch.

Quantity	Mean mismatch	Median mismatch
Amplitude	6.12×10^{-4}	3.73×10^{-4}
Frequency	3.93×10^{-4}	1.73×10^{-4}
h_+	1.64×10^{-2}	4.86×10^{-3}
$h_\times$	1.57×10^{-2}	4.42×10^{-3}

Table 1: Summary of our results. Amplitude and frequency mismatches is calculated with Equation (5), while the polarization mismatches are computed using Equation (7).

For the case of the polarization time series, we calculate the overlap between the original and reconstructed waveform by taking a noise-weighted inner product. The noise-weighted inner product between any two waveform $h_1(t)$ and $h_2(t)$ is given by:

$$\langle h_1|h_2\rangle = 4\,\Re \int_0^\infty \frac{\tilde{h}_1(f)\,\tilde{h}_2^*(f)}{S_n(f)}\,df \tag{6}$$

where, $\tilde{h}_1(f)$ denotes the Fourier transform of $h(t)$ and $\tilde{h}_2^*(f)$ the complex conjugate. $S_n(f)$ is the detector noise, which we take to be given by the `aLIGOZeroDetHighPower` function

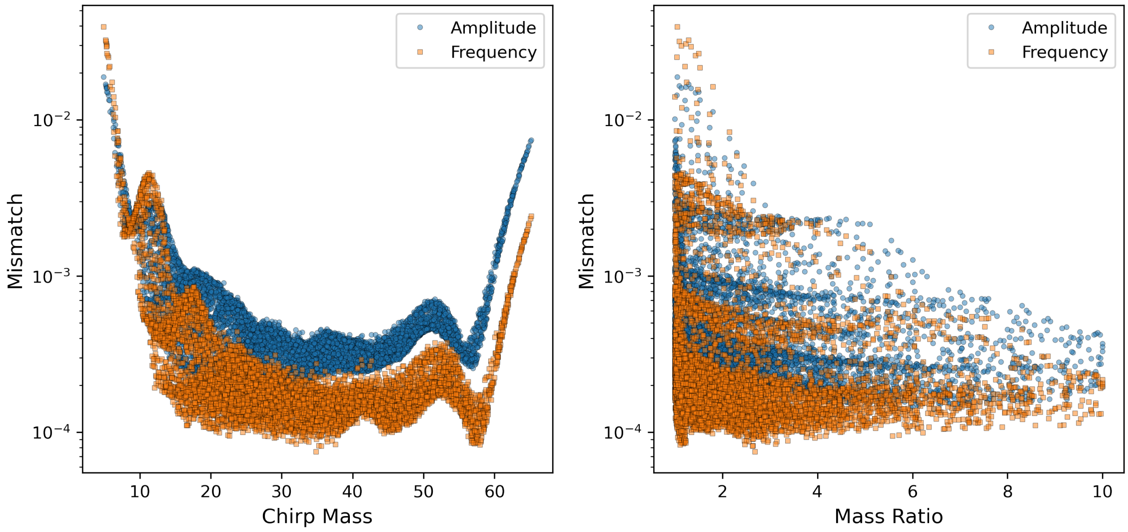

Figure 4: Mismatches calculated for the reconstructed amplitude and frequency time series for all the samples in the test dataset. Left panel shows mismatch variation with the chirp mass, while the right panel is with the mass ratio.

from `pyCBC`. The optimal overlap (faithfulness) and the corresponding polarization mismatch of a reconstructed waveform $\hat{h}(t)$ with the original waveform $h(t)$ can then be calculated as:

$$\tilde{O}(\hat{h}, h) = \max_{\phi_c, t_c} \left[\frac{\langle \hat{h} | h \rangle}{\sqrt{\langle \hat{h} | \hat{h} \rangle \langle h | h \rangle}} \right], \qquad \tilde{\mathtt{MM}}(\hat{h}, h) = 1 - \tilde{O}(\hat{h}, h). \tag{7}$$

The mean and median values of the mismatches $\mathtt{MM}$ and $\tilde{\mathtt{MM}}$ are tabulated in Table 1. Figures 4 and 5 plot these mismatches as a function of chirp mass and mass ratio of the binary source.

5. Conclusion and Futures Prospects

In this paper, we have presented an auto-encoder model for faster generation of effective one-body SEOBNRv4 waveform approximations. To train our model take the time series input polarization waveforms for some set of (m_1, m_2) parameters, convert them to amplitude and frequency series, use optimize the model to accurately generate these back. During inference stage we only pass the parameters as input and obtain the reconstructed polarizations as outputs. Our model is able to generate 10^4 waveforms in O(1) second with a median polarization mismatch of 10^{-3} for the reconstructed waveforms.

Our present work is meant to be a starting prototype model towards the development of a production-ready machine learning based fast and accurate gravitational waveform approximations. Some immediately possible follow-ups in this regards are:

1. Expanding the parameters space to the full 7 intrinsic parameters and including spins, precession, higher-order modes and eccentricity effects in the source binary system.

2. Integrating our model to existing detection and parameter estimation frameworks like `Aframe` [3] and `Bilby` and evaluating the parameter posteriors.

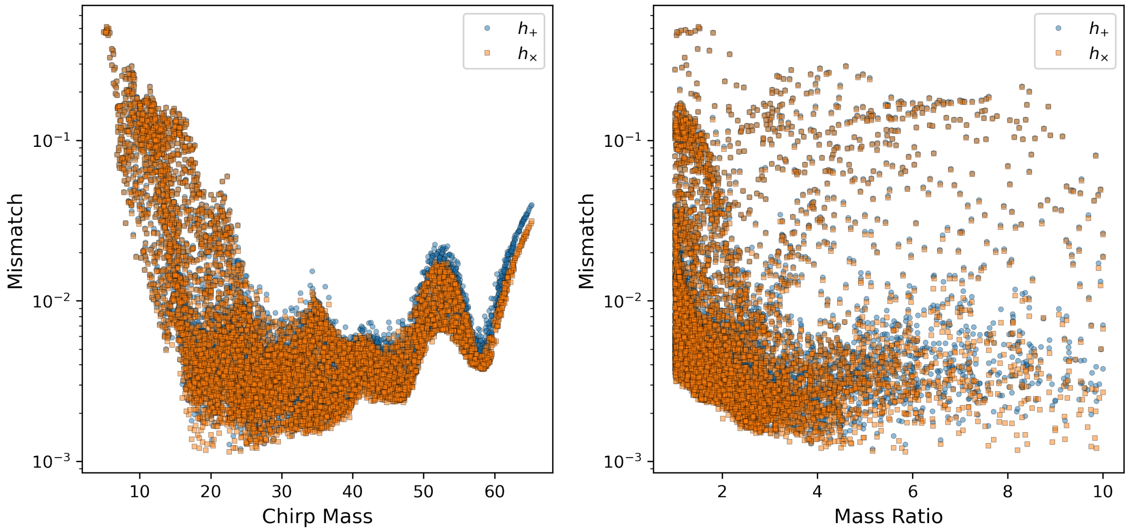

Figure 5: Mismatches calculated for reconstructed polarization time series for all the samples in the test dataset. Layout same as Figure 4.

3. Finishing up and releasing the production-ready full-scale waveform approximation pipeline `umami`, consisting of an ensemble of machine learning models targeting different parts of an IMR waveform and different sub-spaces in the parameter space.

Full scale implementation of our model, a journal paper submission and releases to the wider collaboration are in progress.

Acknowledgements

We would like to thank Kipp Cannon and Feng-Li Lin for insightful discussions. Our model training was performed on CalTech LDG-CIT and UTokyo RESCEU-BBC cluster.

References

[1] E. Cuoco, M. Cavaglià, I. S. Heng, D. Keitel, and C. Messenger. Applications of machine learning in gravitational-wave research with current interferometric detectors. *Living Reviews in Relativity*, 28(1):2, Feb. 2025. ISSN 1433-8351. doi: 10.1007/s41114-024-00055-8.

[2] C.-H. Liao and F.-L. Lin. Deep generative models of gravitational waveforms via conditional autoencoder. *Physical Review D*, 103(12):124051, June 2021. doi: 10.1103/PhysRevD.103.124051.

[3] 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, Mar. 2024.