Deep learning model on gravitational waveforms in merging and ringdown phases of binary black hole coalescences

Joongoo Lee, Sang Hoon Oh, Kyungmin Kim, Gihyuk Cho, John J. Oh, Edwin J. Son, and Hyung Mok Lee

Phys. Rev. D **103**, 123023 — Published 22 June 2021

DOI: 10.1103/PhysRevD.103.123023

Deep Learning Model on Gravitational Waveforms in Merging and Ringdown Phases of Binary Black Hole Coalescences

Joongoo Lee,[1,2,*] Sang Hoon Oh,[3,†] Kyungmin Kim,[1] Gihyuk Cho,[4] John J. Oh,[3] Edwin J. Son,[3] and Hyung Mok Lee[1,2]

[1]*Korea Astronomy and Space Science Institute, 776 Daedeokdae-ro, Yuseong-gu, Daejeon 34055, Republic of Korea*
[2]*Department of Physics and Astronomy, Seoul National University, Seoul 08826, Republic of Korea*
[3]*Division of Basic Researches for Industrial Mathematics, National Institute for Mathematical Sciences, Daejeon 34047, Republic of Korea*
[4]*Deutsches Elektronen-Synchrotron DESY, Notkestrasse 85, 22607 Hamburg, Germany*

The waveform templates of the matched filtering-based gravitational-wave search ought to cover wide range of parameters for the prosperous detection. Numerical relativity (NR) has been widely accepted as the most accurate method for modeling the waveforms. Still, it is well-known that NR typically requires a tremendous amount of computational costs. In this paper, we demonstrate a proof-of-concept of a novel deterministic deep learning (DL) architecture that can generate gravitational waveforms from the merger and ringdown phases of the non-spinning binary black hole coalescence. Our model takes $\mathcal{O}(1)$ seconds for generating approximately 1500 waveforms with a 99.9% match on average to one of the state-of-the-art waveform approximants, the effective-one-body. We also perform matched filtering with the DL-waveforms and find that the waveforms can recover the event time of the injected gravitational-wave signals.

I. INTRODUCTION

Since the first detection of gravitational waves (GW)[1], numerous GW events have been captured by ground-based GW detectors, the Advanced Laser Interferometer Gravitational-wave Observatory (aLIGO) [2] and Virgo [3]. The sources of all events turned out to be compact binary coalescences (CBCs), the collision of two dense objects such as black holes (BH) or neutron stars (NS) — mostly from binary black holes (BBH), 47 out of 50, and partially from binaries containing at least one neutron star [4].

For the type of GW progenitors, template-based GW search is one of the most efficient approaches because the gravitational waveforms from binary mergers can be modeled precisely by multiple methods, e.g., post-Newtonian (PN) for the inspiral phase, numerical relativity for the merger phase, and perturbation theory for the ringdown phase. The template-based search utilizes the matched filtering method [5], which essentially computes the cross-correlation between template waveforms and real GW signal buried in noisy data.

The successful implementation of the matched-filtering-based search relies on the pre-computed waveform templates. Numerical relativity (NR) has been considered as the most accurate method for computing gravitational waveforms. However, obtaining a large number of templates that cover parameter space densely enough for the precise matched filtering search and parameter estimation with NR is not feasible because of too heavy computational requirements. For example, NR simulation of the first GW event GW150914 [1] takes 1-2 weeks using tens to hundreds of CPU cores [6]. In contrast, it takes less than $\mathcal{O}(1)$ seconds to generate inspiral waveforms using post-Newtonian approximations.

Several waveform models approximating NR waveforms have been proposed to reduce the computational cost with reasonable accuracy NR [7–14]. Nonetheless, the physical parameter spaces where each approximant exactly covers are different from each other [12, 15]. Therefore, reserving plural waveform models, complementing each other for various configurations, and saving computing time are crucial for a more elaborate template-based search. It justifies the further study of new waveform approximants.

We present a proof-of-concept demonstration of a deep learning (DL) model for generating gravitational waveforms from the CBC events covering the late phase of inspiral to final ringdown phases. For this purpose, we only consider non-spinning BBH systems for simplicity. Chua et al. [16] utilize deep artificial neural networks to map the physical parameters to coefficients of reduced-order bases waveforms. Williams et al. [17] use Gaussian process regression to approximate the inspiral-merger-ringdown waveforms from the BBH. However, the capability of a fully DL-based deterministic approach has not been explored so far for the generation of the merger-ringdown waveform of CBC[1]. Hence, we examine the viability of the deterministic DL model as a merger-ringdown gravitational waveform model throughout this paper.

While DL models show remarkable performances in a wide variety of fields such as natural language processing (NLP) [18, 19], autonomous driving [20], and image classification [21], most of the models are only capable of handling fixed-size data once they are trained. However, the model we shall adopt for this study should be able to cope with differently-sized data because the length of the waveforms observable by GW detectors depends on the two factors: (1) lower-frequency limit of the detector's sensitivity (around 10 Hz for ground-based detectors) and (2) the masses of the compact binary system [2, 4].

The recurrent neural network (RNN) encoder-decoder-

* ljg4471@gmail.com
† shoh@nims.re.kr

[1] It is known that deterministic models generally show higher accuracy and performance than stochastic methods as the training data is sufficient.

TABLE I. Parameters of the waveform in each dataset. Dataset-1 and-2 have different mass ranges, mass ratios, and numbers of samples, as shown in the table. All the other parameters of both datasets are set to be the same. Note that the waveforms in the datasets are generated in the time domain with `PyCBC` and `SEOBNRv4`.

Variable	Dataset-1	Dataset-2
Mass [min, max]	$[10M_\odot, 40M_\odot]$	$[40M_\odot, 100M_\odot]$
Mass ratio [min, max]	[1, 4]	[1, 2.5]
Number of waveforms(training, validation, test)	(12469, 1533, 1512)	(12447, 1530, 1523)
sampling rate	4096Hz	4096Hz
Distance	100Mpc	100Mpc
Spin	0	0
Inclination angle	30°	30°

based sequence-to-sequence (seq2seq) model [22, 23] designed for NLP is one of the DL models that can handle variable input/output sizes. This model also has shown outstanding performances in many NLP studies [24–27]. The property of gravitational waveforms is similar to that of language type data containing time-ordered words in sentences with different lengths. In that sense, we consider seq2seq as the experimental method to generate waveforms and slightly modify the structure of the model for our purpose.

This paper is organized as follows. Sec. II provides detailed explanations on the data preparation. In Sec III, the original seq2seq model, our modified version, and an evaluation metric for the model performance are elaborated. Sec. IV presents the results of the DL-waveform analysis with GW data and additional dataset-size-associated experiments. Finally, we discuss our results and future research directions in Sec. V.

II. DATA

Since RNN is well-suited to time-series data, we compute non-spinning BBH waveforms in time-domain for training dataset using `PyCBC` [28], a software package for GW data analysis. For this, we use a variant of effective-one-body (EOB) approximants, `SEOBNRv4` [29], one of the most accurate versions of the approximants used in the GW searches.

For the training of the DL model, adopting waveforms obtained by NR is more beneficial than using approximants in the sense of accuracy. However, we find that the number of publicly available NR-waveforms of BBHs is only $\mathcal{O}(10^3)$ [30–33]. In specific, the number of non-spinning BBH waveforms reduces to $\mathcal{O}(10^2)$ [30], so small that it might cause overfitting of the DL model [34], which infects the general performance of the model. Thus we use EOB-waveforms to get a sufficient amount of training samples.

With the software and the approximant, we configure two datasets whose mass ranges of single black holes are $[10M_\odot, 40M_\odot]$ (dataset-1) and $[40M_\odot, 100M_\odot]$ (dataset-2) to divide search regions into low- and high-mass regions. Each dataset is consist of training, validation, and test sub-datasets with respective sample number ratio of 0.8, 0.1, and 0.1. The mass ratios of the sub-datasets are set differently[2]. For the training and validation samples, we use fixed mass ratios with an interval of 0.1 (0.05) within the range of [1, 4] ([1, 2.5]) for dataset-1 (dataset-2). On the other hand, we randomly sample m_1 and m_2 in the corresponding parameter space for the test sub-dataset. In this manner, we can prove that the model trained with a limited mass ratio samples can be applied to the ones residing in any regions of the parameter space. Fig. 1 shows the scatter plots of m_1 and m_2 of training sub-dataset in dataset-1 and -2 with color-coded chirp masses defined as $M_{ch} = (m_1 m_2)^{3/5}(m_1+m_2)^{-1/5}$. We use the sampling rate,

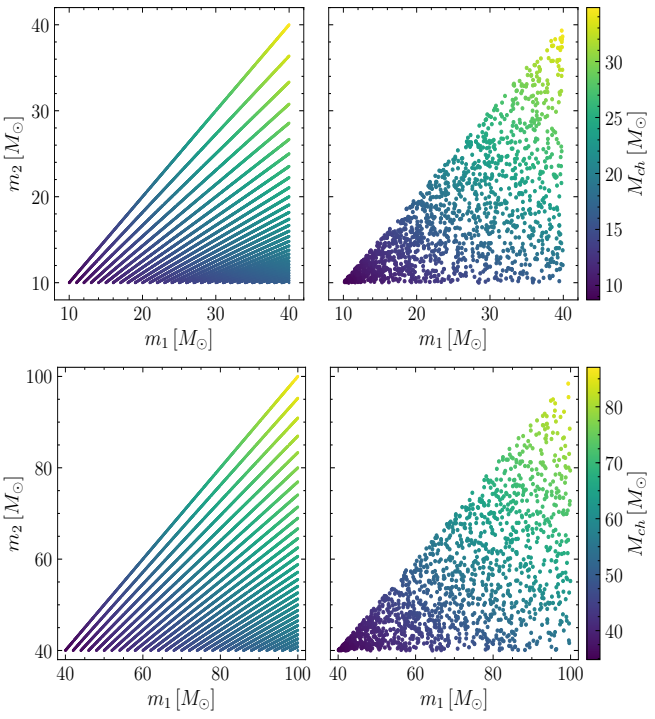


FIG. 1. The component masses of training (left) and test (right) sub-datasets in dataset-1 (upper) and dataset-2 (bottom) with the color-coded chirp mass. While we use a set of fixed mass ratios, m_1/m_2, for the training sub-dataset, m_1 and m_2 are randomly chosen for the test sub-dataset with the restriction that $m_1 \geq m_2$. The mass ratios range from 1 to 4 for the dataset-1 and from 1 to 2.5 for dataset-2.

[2] The mass ratio is defined as m_1/m_2, and $m_1 \geq m_2$ is assumed by convention.

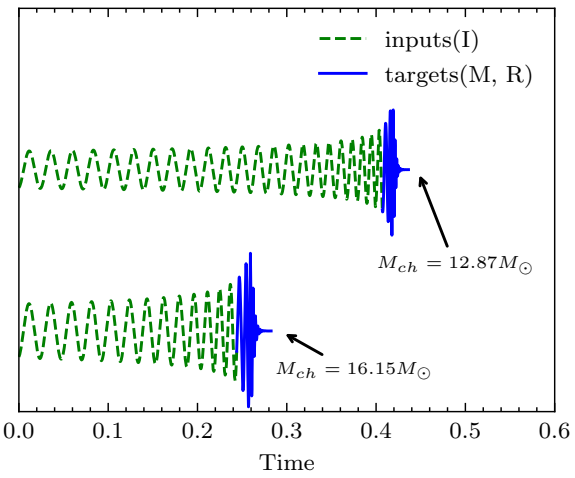


FIG. 2. Examples of input (green dashed; inspiral) and target (blue solid; merger-ringdown) waveforms drawn with different chirp masses of the compact binary system. They are computed by using `SEOBNRv4`. The upper and lower waveforms depict $M_{ch} = 12.87M_\odot$ and $M_{ch} = 16.15M_\odot$, respectively. Note that the length of the generated waveforms changes depending on the mass.

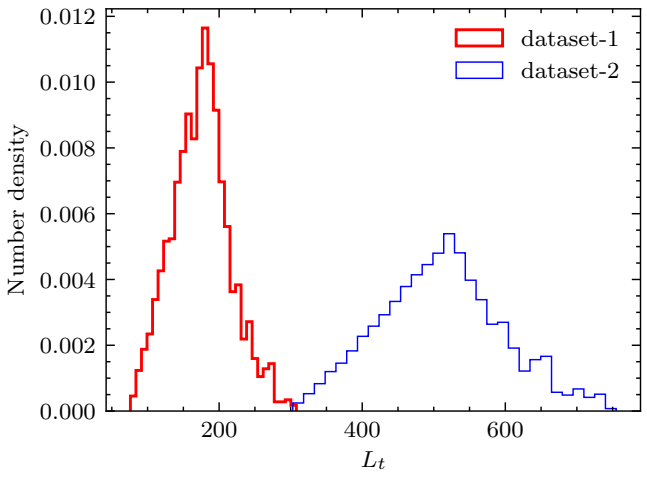


FIG. 3. target waveform length (L_t) distribution of the training sub-dataset in dataset-1 (thick red) and dataset-2 (thin blue). Note that the non-uniform distributions are caused by the parameter sampling and input-target split method described in Sec. II.

distance, and inclination angle of 4096Hz, 100Mpc, and 30°, respectively. The parameters employed for data preparation are tabulated in Table I.

Following the data generation, the waveforms in dataset-1 and -2 are normalized with the maximum strain amplitude of each dataset. Since the diverse range of samples may cause biased results [35], data normalization for the differently ranged dataset is necessary. By normalizing the dataset, the sample values can be restricted in a comparable range and contribute equally to the DL model optimization at the beginning of the training.

In turn, we divide each waveform into input and target waveforms: the input with the inspiral phase and target with merger and ringdown phases, respectively. For the division, we consider the point that the GW frequency reaches the innermost stable circular orbit (ISCO) frequency [36] as the termination point of the inspiral phase [37]. The final data point of the input waveform and the initial data point of the target waveform are intentionally superposed to check whether the DL-waveform and given inspiral waveform are smoothly connected. Fig. 2 illustrates examples of input and target waveforms with different chirp masses. For the training of our DL model, we feed the input waveform to the DL model and let the model recover target waveform.

For divided target waveforms, we illustrate the number density distributions of waveform lengths in Fig. 3 (denoted by L_t). As shown in the figure, the distributions are not uniform. We reckon that this non-uniformity causes L_t-dependent accuracy of the DL model, which will be discussed in Sec. IV A.

III. METHODS

Since the duration of the GW emission within the detector's sensitive frequency band varies depending on the component masses or chirp mass of the binary system, we need a DL model capable of handling different size data. For this, we design a DL model with a novel architecture based on seq2seq, which is built for NLP. In this section, we briefly overview the original seq2seq model[3] and elaborate on our model below.

A. Original Sequence-to-Sequence Model

DL models for NLP take a batch of sentences as inputs and output transformed sentences. For that, each word in the sentences should be digitized since machine learning models work numerically. With the linguistic property that the number of vocabularies in a specific language is limited to a finite number, each distinct word can be represented as a vector by word embedding [38]. Thus, the sentence prediction problem can be regarded as selecting words from a given dictionary. The vectorized sentences, however, have different sizes because every sentence is composed of a different number of words.

To resolve the issue, the encoder, mapping the variable size input sequence into a fixed-size vector, is employed in the seq2seq model. Afterward, the transformed vectors, so-called representations, by the encoder are transmitted to the decoder, and it sequentially recovers the variable size target sentences. In the decoder calculation process, the output at the previous computing-step is taken as the input of the next step.

[3] For more details of the original model, we refer to [22, 23].

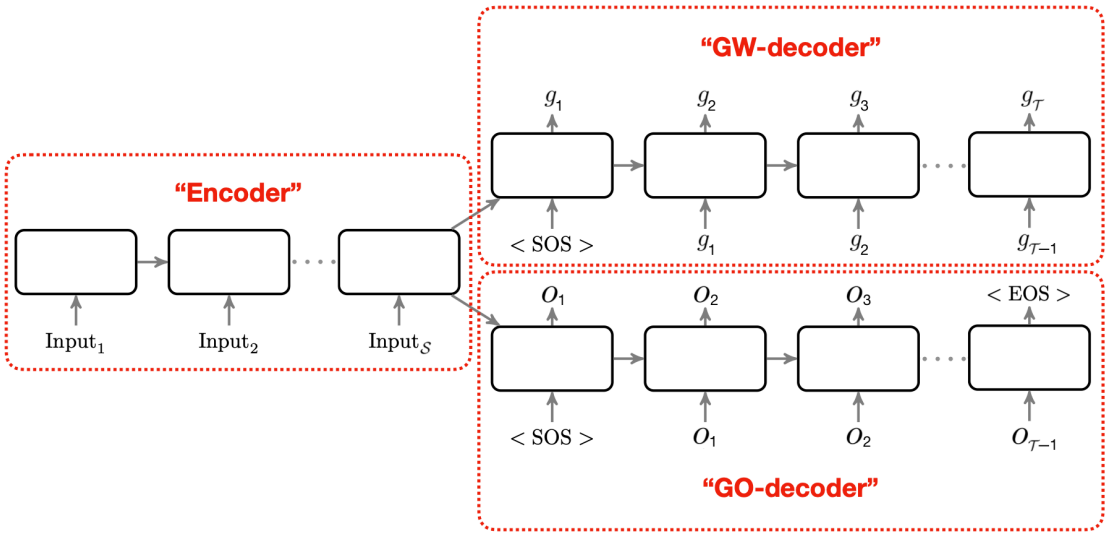


FIG. 4. The schematic workflow of the DDS2S model. The solid black boxes indicate RNN cells. The model sequentially takes $\mathcal{S}$ vectors as input waveforms and attempts to regenerate target waveforms and GO-function, $\mathcal{G}$. The decoders start computation when inputted $\langle$SOS$\rangle$ and retrieve $\mathcal{T}$ vectors as output waveforms until the GO-decoder yields a value under 0.5, marked by $\langle$EOS$\rangle$. Note that the decoders use the output of the previous computing-step as the input at the next computing-step. The detailed structural information of the model is tabulated in Table. II.

Each sentence is required to end with the end-of-sequence token ($\langle$EOS$\rangle$), and the decoder starts and finishes its computation by taking and outputting $\langle$EOS$\rangle$. The conditional vector $\langle$EOS$\rangle$ can be defined differently depending on the user's preference.

B. Dual-Decoder Sequence-to-Sequence Model

In the work of the original model, Sutskever et al. [23] were able to construct the $\langle$EOS$\rangle$, the interrupting condition of the decoder computation, using the linguistic property that the number of vocabularies is limited. Since the words in the dictionary can be discretely distinguished, it is clear to set the condition.

Regarding the GW-data, however, it becomes hazy to establish a criterion for interrupting the computing-step because the strain amplitudes of GWs are continuous real numbers: the number of possible cases is infinite, unlike the words in a dictionary. Thus, we cannot expect the model to produce an output that exactly matches a specific number by all digits. For example, when we set $\langle$EOS$\rangle = 0$, the model is unlikely to obtain the exactly matching value in machine precision.

As a strategy for learning this continuous sequence, we design a modified seq2seq model (DDS2S, Fig. 4) with one encoder and dual-decoder, GW- and GO-decoder: the encoder encrypts input waveforms, GW-decoder recovers target waveforms, and GO-decoder predicts the length of the target waveforms. While the computational mechanisms of the encoder and decoders are identical to the ones in the original model, the approach for handling input and target data is different.

First, the input and target waveforms are divided into the number of $\mathcal{S}$ and $\mathcal{T}$ vectors with $\mathcal{R}$ elements. When $\mathcal{R} > 1$, the ends of the waveform elements are zero-padded before division to match the component numbers with the multiples of $\mathcal{R}$. The zero-padded lengths of input and target waveforms can be computed via $L_s = \mathcal{S}\mathcal{R}$ and $L_t = \mathcal{T}\mathcal{R}$ [4]. Then, the encoder sequentially takes $\mathcal{R}$ elements of input waveforms $\mathcal{S}$ times and encrypts them into fixed-size vectors. The encoder outputs are transmitted to GW- and GO-decoders.

Subsequently, the GW-decoder regenerates $\mathcal{R}$ elements of target waveforms at every computing-step throughout the $\mathcal{T}$ step[5]. The generated waveforms are stacked in the order of computing-step and compared with the target waveforms to calculate the error function. As the error function of the GW-decoder, $\mathcal{I}$, we use the sum of mean-squared error and negative cosine similarity;

$$\mathcal{I}(g,t) = \frac{1}{\mathcal{T}}\Sigma_i(g_i - t_i)^2 - \frac{g \cdot t}{||g||\,||t||}, \tag{1}$$

where g and t are respectively the generated and target wave-

[4] Note that L_s and L_t are the lengths of input and target waveforms without zero-padding as $\mathcal{R} = 1$.

[5] The total computing-step multiplied by $\mathcal{R}$ and waveform length are compatible concepts, and one can convert them into the duration of GW by multiplying the inverse of the sampling rate, 4096Hz.

forms; $\mathcal{T}$ is the number of vectors for the given target waveform; $||\cdot||$ is L^2 norm.

Lastly, we establish the GO-function to endow the GO-decoder the capability to estimate the length of the target waveform. When the given target waveform consists of $\mathcal{T}$ vectors, we can set the integer condition, $\mathcal{C}$, for progressing from computing-step τ to $\tau+1$ as follows: 1 for proceeding and 0 for breaking.

$$\mathcal{C}_\tau = \begin{cases} 1, & \text{if} \quad 1 \leq \tau < \mathcal{T}-1 \\ 0, & \text{if} \quad \tau \geq \mathcal{T}. \end{cases} \tag{2}$$

We may use the set of $\mathcal{C}_\tau$ to train GO-decoder, but the discrete values and rapid decrease of $\mathcal{C}$ from $\tau = \mathcal{T}-1$ to $\tau = \mathcal{T}$ are inappropriate for the training of the DL model. Thereby, we define GO-function, $\mathcal{G}$, approximating the integer $\mathcal{C}$ values with a smooth decreasing pattern near $\tau = \mathcal{T}$ and use the function to compute the mean-squared error with the GO-decoder outputs. The GO-function and the error function, $\mathcal{J}$, of the GO-decoder are described below.

$$\mathcal{G}_\tau = \begin{cases} 1 - 0.5\,(\tau/\mathcal{T})^\alpha\,, & \text{if} \quad 1 \leq \tau \leq \mathcal{T}-1 \\ 0, & \text{if} \quad \tau \geq \mathcal{T}, \end{cases} \tag{3}$$

$$\mathcal{J}(o,\mathcal{G}) = \frac{1}{\mathcal{T}}\Sigma_i (o_i - \mathcal{G}_i)^2, \tag{4}$$

where o_i is the output of GO-decoder. Fig. 5 presents how the curve of the $\mathcal{G}$ varies according to different αs. As the α is getting bigger, the GO-function approximates the $\mathcal{C}$ values more accurately. On the contrary, we find that the rapid decrease of $\mathcal{G}$ near $\tau = \mathcal{T}$ hinders the training of the DL model when the α is too high. We empirically determine α of 5 for the training of the model.

The final loss for the training is the sum of the error function of GW- and GO-decoders, namely $\mathcal{I} + \mathcal{J}$. The model is trained by adjusting its parameters in such a way the error is minimized.

We apply the `Sigmoid` to the output layer of the GO-decoder since the GO-function should output values from 0 to 1. Then, we have given output values rounded to either 0 or 1. The computation continues when the rounded value is 1 and stops otherwise. Hence, the GO-decoder output below 0.5 serves as $\langle$EOS$\rangle$ in our case. For this reason, we define $\mathcal{G}$ to have a slightly higher value than 0.5 at $\tau = \mathcal{T}-1$ because we expect the model to stop calculating at $\tau = \mathcal{T}$. For the DDS2S model, we newly define zero vectors with $\mathcal{R}$ elements as a start-of-sequence token ($\langle$SOS$\rangle$), which is inputted at the start of decoder computation.

Among the prominent RNN cells, we choose Gated Recurrent Unit (GRU) [22] for the encoder and both decoders because the setting with GRU showed higher accuracy and faster training than Long-Short Term Memory [39, 40], another well-known RNN cell. A fully connected layer is placed at the end of the decoders' hidden layers to convert hidden states to vectorized outputs with $\mathcal{R}$ components. We use the hyperbolic tangent as the activation function for hidden layers of each RNN cell of encoder and decoders.

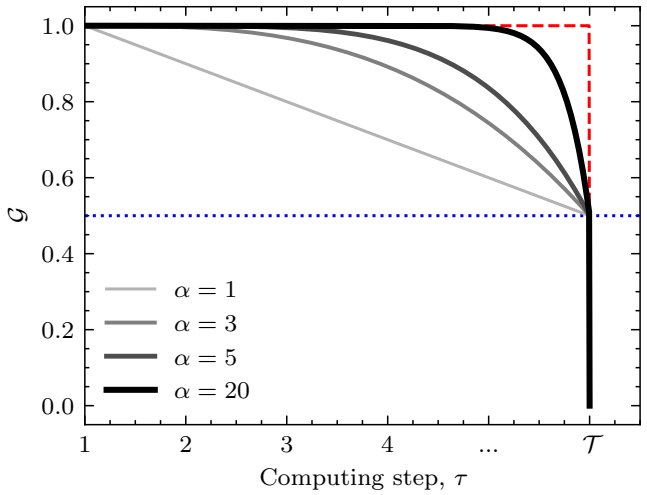


FIG. 5. The GO-function, $\mathcal{G}$, with several values of α in greyscale. The red dashed-line depicts how the integer condition, $\mathcal{C}$, changes according to the computing-step. As the value of the α increases, the function approximates the $\mathcal{C}$ values more accurately. We also draw the horizontal blue dotted-line at 0.5, the condition of interrupting decoders' computation.

For the model structure, we find an empirically optimal model configuration varying the number of neurons in hidden layers (hereafter, hidden neurons) based on the overlap to a reference waveform, which we will discuss in the following sub-section. The information on the network configurations and hyperparameters of the optimal model is summarized in Table II.

C. Overlap

We use overlap to assess the DL-waveforms' accuracy. The normalized overlap, $\mathcal{M}$, of the DL-waveform g and the target t can be computed as

$$\mathcal{M} \equiv \frac{(g|t)}{\sqrt{(g|g)(t|t)}}, \tag{5}$$

where $(g|t) = \int_{-\infty}^{\infty} \tilde{g}(f)\tilde{t}^*(f)df$. $\tilde{g}$ and $\tilde{t}$ are the Fourier transform of g and t, respectively, and asterisk mark (*) is complex conjugate. Note that $\mathcal{M}$ becomes 1 for the perfect match and 0 for the perfect mismatch between g and t.

From the grid-search described in Appendix A, we choose an empirically optimal model configuration, maximizing the minimum overlap of the model's output waveforms. Providing accuracy, we use the setup with 256 hidden neurons and $\mathcal{R} = 1$. Henceforward, we shall only discuss the results of the model with 256 hidden neurons and $\mathcal{R} = 1$. The detailed explanation can be found in Appendices A and B.

TABLE II. Detailed structure of the DDS2S model.

	Encoder	GW-Decoder	GO-Decoder
RNN cells	GRU	GRU	GRU
The number of RNN cells	$\mathcal{S}$	$\mathcal{T}$	$\mathcal{T}$
The number of input layers	1	1	1
The number of hidden layers	4	4	4
The number of output layers	-	1	1
The number of input neurons	$\mathcal{R}$	$\mathcal{R}$	1
The number of hidden neurons	256	256	256
The number of output neurons	-	$\mathcal{R}$	1
Activation function of input layers	`Tanh`	`Tanh`	`Tanh`
Activation function of hidden layers	`Tanh`	`Tanh`	`Tanh`
Activation function of output layers	-	-	`Sigmoid`

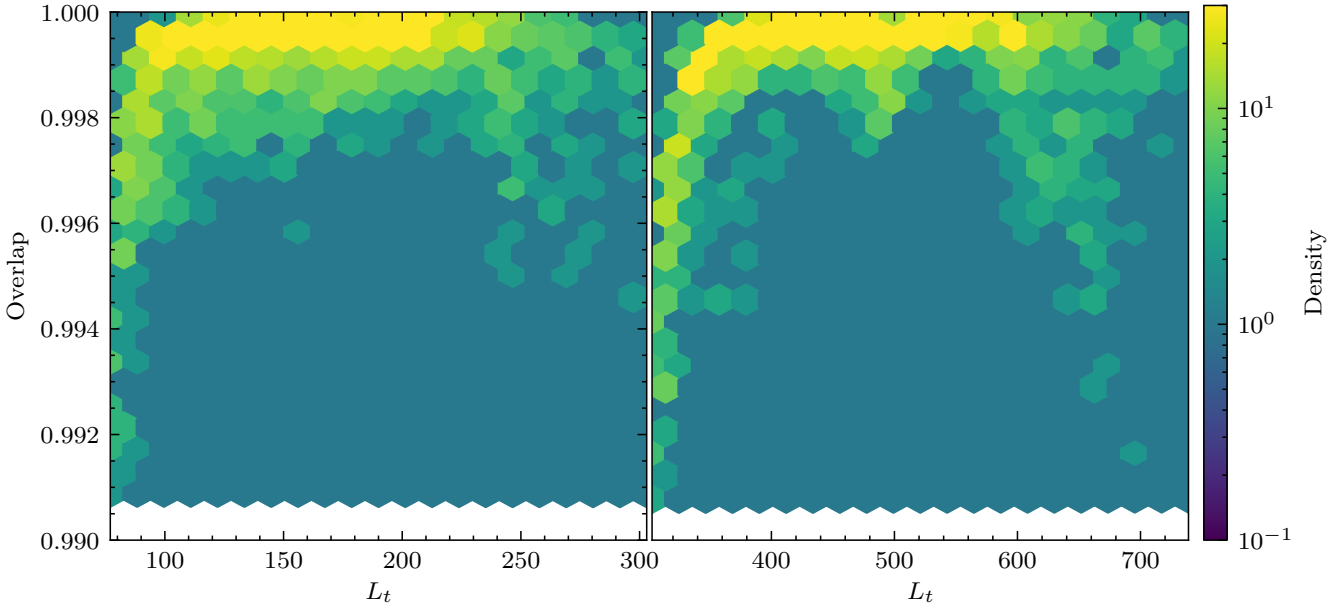


FIG. 6. Density heatmap of overlap according to target waveform lengths, L_t, for the dataset-1 (left) and 2 (right). We draw the vertical axes of the two plots in the same range and scale. For clear contrast, we leave the regions with no samples empty at the bottom of the plots. As shown in the plots, overlaps of all the DL-waveforms are higher than 0.990. Besides, the averages of the waveforms from both datasets are over 0.999. However, a few shortest and longest samples have smaller overlap values. Considering the relatively small number of the shortest and longest waveforms in the training sub-dataset (Fig. 3), it implies that the non-uniformity of the sub-dataset is related to the locally different accuracy of the DL model.

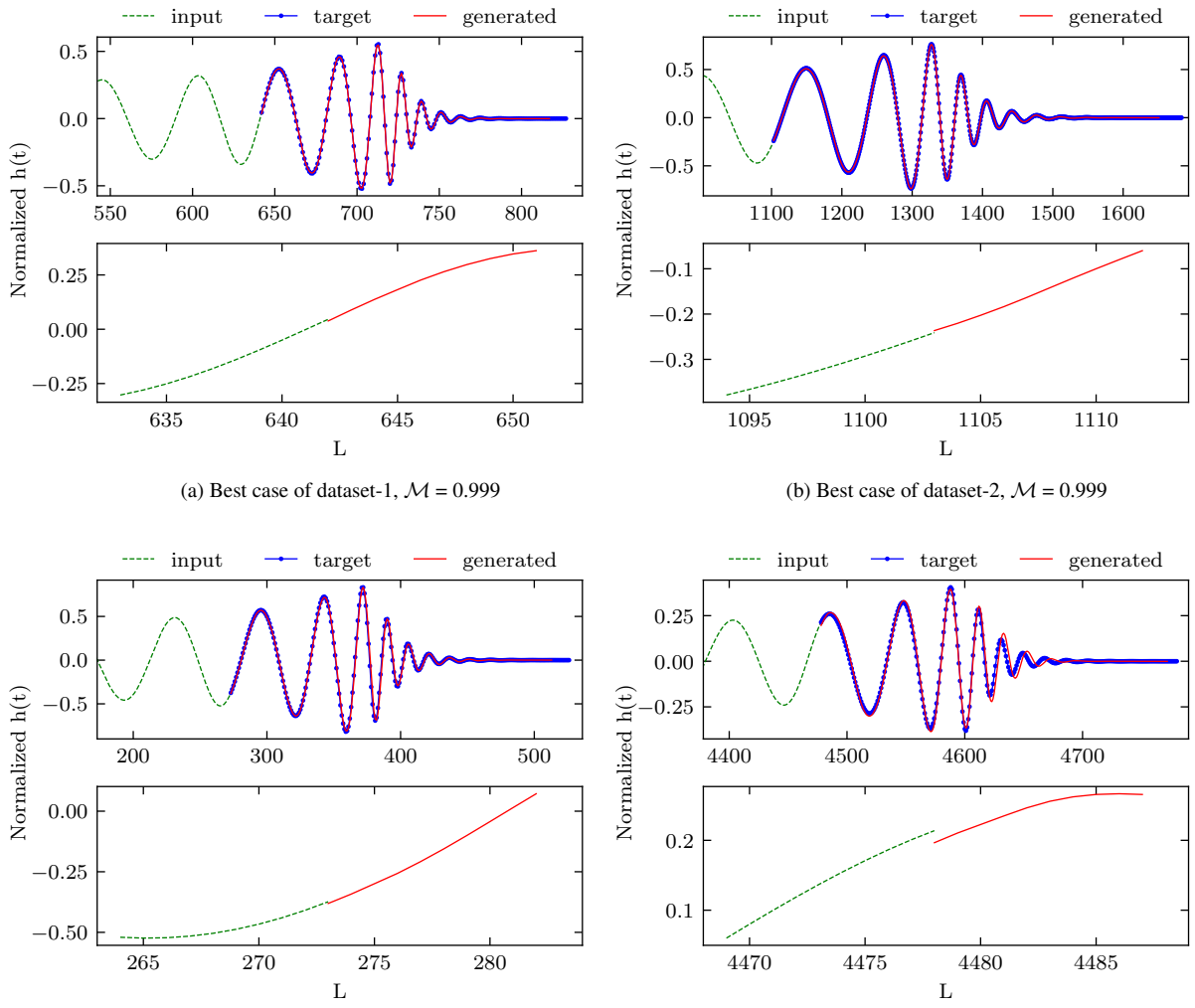
IV. RESULTS

A. Waveform Validation

The Fig. 6 depicts the overlap density heatmap between the DL-waveforms and corresponding target EOB-waveforms of the dataset-1 and 2. All of the DL-waveforms are in excellent agreement with their target waveforms in both cases as the minimum value of overlaps is higher than 0.990[6]. Furthermore, the mean overlaps of waveforms from both datasets are higher than 0.999, indicating less than 0.1% average error.

However, as we can see from the figure there are several outliers whose overlaps are substantially smaller than the majority. We explore the dependence of the overlap on the target waveform length to track down possible reasons for relatively poor overlap cases. The heatmap shows the distribution of the overlaps concerning the length of the target waveforms. The overlap of dataset-1 (dataset-2) tends to decrease at the short-end and long-end of the target waveform length, i.e., $L_\mathrm{t} \lesssim 100$ or $L_\mathrm{t} \gtrsim 250$ ($L_\mathrm{t} \lesssim 400$ or $L_\mathrm{t} \gtrsim 600$). As shown in Fig. 3, the training samples in the range of $100 \lesssim L_\mathrm{t} \lesssim 250$ of dataset-1 and $400 \lesssim L_\mathrm{t} \lesssim 600$ of dataset-2 dominate the number distribution of the target waveform length. It can be attributed to the fact that the model is more likely to weigh the majority of

[6] For comparison, the authors of Ref. [41] have shown that the overlap between numerical and their phenomenological waveforms ranges from 0.95 to 0.99. On the other hand, Ref. [42] have shown their model results in the overlap ≥ 0.99.



(a) Best case of dataset-1, $\mathcal{M} = 0.999$

(b) Best case of dataset-2, $\mathcal{M} = 0.999$

(c) Worst case of dataset-1, $\mathcal{M} = 0.991$

(d) Worst case of dataset-2, $\mathcal{M} = 0.991$

FIG. 7. The input (green dashed), target (blue solid line with dots), and DL- (red solid) waveforms from dataset-1 (left column) and dataset-2 (right column) with the amplified images of connection points. The horizontal and vertical axes indicate the length of the waveforms in sampling unit and the normalized strain amplitude of the GWs, respectively. We only show a hundred sampling units of input waveforms in the plots for clear visualization. The top and bottom panels are the waveforms with the highest and lowest overlap cases, respectively.

the training sub-dataset.

We also visually inspect the agreement between the DL-waveforms and target waveforms. Fig. 7 shows the best and worst overlap cases of the DL-waveforms. The overlaps of the best cases for both datasets are $\mathcal{M} = 0.999$. The time-series of the DL-waveforms matches well with the target waveforms. For the worst cases, the overlaps of the two datasets are both 0.991 (Fig. 7 (c) and (d)). We see that there exist small discontinuities between the DL- and input waveforms as shown in the lower panel of the figure. We may resolve the discontinuity by post-processing or letting the DL model generate the whole waveform in the inspiral-merger-ringdown phase at once. We leave this issue to future work.

B. Injection Test

Next, we attempt to use the DL-waveform templates in simplistic search of parameters, i.e., m_1, m_2, and the event time of the simulated GW signals. To replicate practically used waveform templates, we hybridize inspiral `SEOBNRv4`-waveform and merger-ringdown DL-waveform by simply concatenating the two waveforms. One may implement sophisticate hybridization of waveforms, but it is beyond the scope of this work. We perform parameter grid-search instead of Markov Chain Monte Carlo, typically executed for the parameter estimation of GWs [43], due to the practical difficulty of plugging a new waveform model in the existing parameter estimation code [44]. For the computation of SNR and the search of the events, the matched filtering engine of `PyCBC` [45] is used.

To simulate the observation data embracing a GW signal, we use the LIGO-Hanford O1 data provided by GW Open Science Center[7]. We randomly select a 32-second segment from the data without any known GW signals and inject a SEOBNRv4-waveform into the center. While we use five sets of different injection parameters and distances, we fix the inclination angle to 30°for simplicity. The configuration setups of the tests are tabulated in the first three columns of Table III.

By performing the parameter grid-search for multiple injection waveforms, we retrieve injection parameters in all examinations within the 90% confidence interval. We first define the search parameter sets, (m_1, m_2) on regularly-spaced grid of the parameter space. Then, we construct the full IMR waveform templates by hybridizing the inspiral waveform and the merger-ringdown DL-waveform using SEOBNRv4 and DDS2S, respectively, for the parameter sets. Across the parameter sets, we compute SNR by matched filtering with each waveform template using PyCBC on the simulated data. Assuming the likelihoods of the parameter sets are proportional to the SNR, we estimate the probability density function (PDF) of the parameters. Then, we marginalize the PDF with respect to each parameter and acquire the median as the best-fit parameters with their 90% confidence interval. Subsequently, we repeat the entire process with different combinations of injection masses and distances. The best-fit parameters with confidence intervals and their SNRs are summarized in the last two columns of Table III.

The best-fit parameters and the high SNR region emerge around the chirp mass contour line of the injected signal. Since the chirp mass of GW is governed by the frequency and frequency derivative [46], and its SNR depends on frequency evolution [47], the SNR of GW again relies on the chirp mass. It is well-reflected in the example contour map of the signal with $m_1 = 35M_\odot$ and $m_2 = 20M_\odot$ (Fig. 8).

Using the best-fit parameters found from the grid search, we perform event time searches and find the SNR peak at where we inject the signals. We illustrate SNR time-series of the above example case in Fig. 9. As can be seen in the figure, the peak SNR occurs at the center of the data segment, where we have injected the simulated signal.

It is known that the systematic error from waveform approximants is independent of SNR, while the statistical error due to noise roughly scales as $1/\mathrm{SNR}$. One can readily expect that the systematic error could dominate in higher SNR signals. Cutler and Vallisneri [48] have presented rigorous computation of the systematic errors in parameter estimation using 3.5PN (post-Newtonian approximation of order 3.5) waveforms for inspiral signals of massive black hole binaries. They have shown that the magnitude of the systematic errors from 3.5PN waveforms with $\mathcal{M} > 0.9999$ commensurate with the SNR ~ 1000 statistical errors. Motivated by this, we roughly estimate the impact of systematic error of our DL-based waveform on the parameter estimation by repeating the grid-search of parameters as described above with varying SNR of the injected signal. By comparing the systematic error with the statistical errors of the same parameter as increasing the SNR of the injected signal, we find that the magnitude of the systematic error becomes comparable to the 1-σ statistical error at SNR $\sim \mathcal{O}(10)$ in our DL-based waveform approximant.[8]

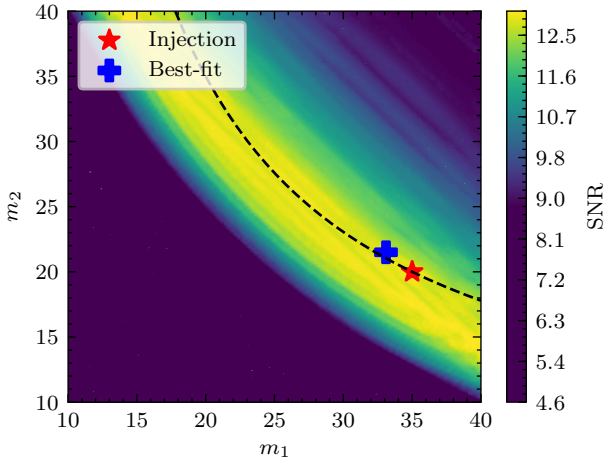


FIG. 8. Filled contour map of SNR in the parameter space for the injection signal with $m_1 = 35M_\odot$ and $m_2 = 20M_\odot$. Each of the red star and blue plus markers indicates injection and best-fit parameters. The black dashed line is a contour with the level of injection chirp mass. The best-fit parameters and the high SNR region arise in the vicinity of the contour line. Although our parameter space is restricted with the condition $m_1 \geq m_2$, the filled contour map is reflected on the slope of 1 line for aesthetic visualization.

C. Performance Dependence on the Dataset Size

We inspect the dependence between the accuracy of the DL model and the number of waveforms in the training subdataset. The test is performed to explore the viability of applying the proposed model to NR-waveforms, in which only a few thousands exist [30–33]. We generate four reduced datasets with half and the tenth number of waveforms in the original training data of dataset-1 and -2, maintaining the number of waveforms in the validation and test data.

We find that one-tenth of the original size is enough to reach the required accuracy of $\mathcal{M} \geq 0.99$. The model is trained more than five times with each reduced training data. It turns out that the minimum and average values of overlap are higher than 0.990 and 0.999, equivalent to 1.0% and 0.1% error, respectively, for all DL-waveforms of the trained model from each run. The mean values for the averaged overlaps and minimum overlaps from more than five individual runs are tabulated in Table IV. We also present the results of Sec. IV A for comparison in the last column. The relative dataset size in the

[7] https://www.gw-openscience.org/archive/O1/

[8] Note that our approach for finding the SNR level where the two errors become similar is not rigorous. For a more in-depth exploration of the systematic errors, refer to [48].

TABLE III. Summarized results of the injection tests. The best-fit parameters and their SNR for the injected signals are computed by PyCBC matched filtering engine with DL waveform templates. We establish template waveforms by hybridizing inspiral SEOBNRv4 and merger-ringdown DL waveforms. The m_1 and m_2 are given in the unit of the solar mass. I, M, and R indicate inspiral, merger, and ringdown phases, respectively.

Template approximant	Distance (Gpc)	Injection (m_1, m_2)	Best-fit (m_1, m_2)	SNR
EOB (I) + DL (MR)	1.6	80.0, 65.0	$80.1^{+13.7}_{-14.5}$, $61.7^{+18.3}_{-16.4}$	14.5
	1.5	70.0, 60.0	$73.9^{+16.5}_{-16.9}$, $58.6^{+16.6}_{-14.4}$	13.0
	0.8	35.0, 20.0	$33.1^{+5.6}_{-6.8}$, $21.5^{+9.0}_{-8.4}$	12.7
	0.7	30.0, 25.0	$31.6^{+6.3}_{-7.0}$, $22.7^{+8.6}_{-8.3}$	15.3
	0.6	25.0, 20.0	$28.3^{+8.0}_{-8.4}$, $18.9^{+7.1}_{-6.6}$	15.7

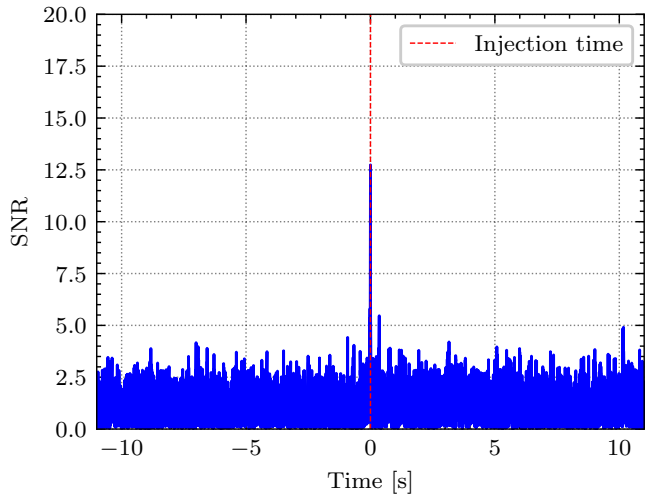


FIG. 9. SNR time-series computed by matched filtering engine of PyCBC and best-fit DL-waveform template of Fig. 8. The injected signal is the SEOBNRv4-waveform ($m_1 = 35M_\odot$ and $m_2 = 20M_\odot$). Here, we initialize the start time of the injected signal to 0, marked by the red dashed line. Note that the SNR peak occurs at the injection time.

TABLE IV. Accuracy variation of the DL model according to dataset size. We also show the results of Sec. IV A in the last column of the table for comparison. The mean values for the minimum and average overlaps from more than five individual runs for each dataset are summarized in the table. The value of the relative dataset size is the ratio of the number of waveforms between the reduced training sub-dataset and the sub-dataset introduced in Sec. II.

Relative dataset size	0.1	0.5	1
Minimum overlap (dataset-1)	0.991	0.990	0.991
Minimum overlap (dataset-2)	0.990	0.990	0.991
Average overlap (dataset-1)	0.999	0.999	0.999
Average overlap (dataset-2)	0.999	0.999	0.999

table means the ratio of the number of waveforms in the training data to the number of waveforms in the original training data. The result shows that reducing the number of waveforms down to 1000 for the training hardly affects obtaining the desired accuracy. Hence, we advocate that the application of the DDS2S model to NR-waveforms is feasible.

V. SUMMARY AND DISCUSSION

The efficiency of matched filtering for searching GW signals buried in noisy GW data has been proved by recent successful detections of GW signals. Although NR can increase the accuracy of template waveforms, expensive computational costs of running NR limit the use of it for the generation of a sufficiently large number of template waveforms. This drawback of NR eventually led to the use of approximate waveforms for the matched filtering instead. Motivated by such difficulties, we have examined the DL method for the generation of template waveforms with much smaller computational costs but comparable accuracy to NR.

To study the feasibility of this consideration, we have implemented the DDS2S model. The encoder-decoder structure is capable of handling the variable sizes of different waveforms, and the dual-decoder structure enables the model to control the continuous real-numbered sequences.

We also have examined the applicability of the waveforms by computing the overlap with EOBNR-based waveforms and performing the injection test. The accuracy of the DL-based waveforms is found to be better than 99.9 % in most combinations of the masses, while a small number of outliers with overlap as small as 0.99 exists. In the injection test, we have recovered the event time of waveforms injected into real noise data with the conventional matched filtering engine of PyCBC.

We have found that the method generating merger-ringdown waveforms using the inspiral waveforms needs to be improved. For example, we have seen that discontinuities occurred between input and output waveforms, as shown in Fig. 7, although the minimum overlap of DL-waveforms to the EOB-waveform was higher than 0.990. To avoid this issue, we may take a new strategy of generating a full IMR waveform. However, the main goal of this paper is to demonstrate the feasibility of adopting DL to model the merger-ringdown waveforms. Hence, we leave the implementation of a DL model generating the full waveforms to future work.

Regarding the speed of waveform generation, the DDS2S model has an advantage over other waveform approximants when computing a batch of multiple waveforms simultaneously. For computing a single waveform, EOB is faster than the DDS2S model, typically taking $\mathcal{O}(10^{-2})$ seconds using a modern CPU core. However, the DDS2S model generates $\sim$ 1500 waveforms using pre-generated inspiral waveforms

in $\mathcal{O}(1)$ seconds using NVIDIA GeForce GTX 1080, while EOB took $\mathcal{O}(10)$ seconds. The disparity arises since the DL models are specialized for batch computations, which process multiple data at once.

The DDS2S model has been built to learn how to predict the output waveforms only from the given input waveforms without any specific physical information of the source binary system. Thus, we can readily extend this work to various systems of interest.

For a more precise description of realistic physical binary systems, we need to have waveform models for more complex binaries: a wider range of the mass ratios, the spin of each component, eccentricity of the orbits. GWs from unbound orbit such as hyperbolic and parabolic encounters are also of great interest. Lastly, it is worthwhile to mention that recalibration of full IMR waveforms to increased amounts of NR waveform data is in progress in the community. [49]

Our approach described in this paper can potentially be applied to more complex systems described above because DDS2S only depends on training data, not any assumptions or approximations on which other waveform models are based. Moreover, we have observed that ~ 1000 training waveforms are sufficient for the model to reach the expected level of accuracy in Sec. IV C. Thus, as long as there is a sufficiently large number of training waveform samples for any systems or NR are given, DDS2S can be trained to generate accurate waveforms in principle.

ACKNOWLEDGMENTS

The authors are grateful to Chunglee Kim and Hyung Won Lee for their helpful comments. And the authors also thank Young-Min Kim, Hee-Suk Cho, and Whansun Kim for fruitfully discussion. G.C is supported by the ERC Consolidator Grant "Precision Gravity: From the LHC to LISA" provided by the European Research Council (ERC) under the European Union's H2020 research and innovation programme, grant agreement No. 817791. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1A2C2006787). This work is partially supported by the Global Science experimental Data hub Center (GSDC) at KISTI. This research was also partially supported by National Institute for Mathematical Sciences (NIMS) funded by MSIT (B19720000) and the NRF grant funded by the MSIT (NRF-2020R1C1C1005863, NRF-2020R1I1A2054376). This research has made use of data, software and/or web tools obtained from the Gravitational Wave Open Science Center (https://www.gw-openscience.org), a service of LIGO Laboratory, the LIGO Scientific Collaboration and the Virgo Collaboration. LIGO is funded by the U.S. National Science Foundation. Virgo is funded by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale della Fisica Nucleare (INFN) and the Dutch Nikhef, with contributions by Polish and Hungarian institutes. This paper has the LIGO document number LIGO-P1900207.

Appendix A: Empirically Optimal Number of Hidden Neurons

We investigate the influence of the hidden neurons on the accuracy of the models; 64, 128, and 256 hidden neurons.

Accuracy-wisely, we find that the model with 256 hidden neurons is most suitable amid the tested cases. To compare model accuracy according to the number of hidden neurons, minimum and average overlap between DL-waveforms and corresponding target waveforms are computed. Table A.1 summarizes the minimum and average overlaps of the models for dataset-1 and -2. The minimum overlap values of each model from dataset-1 (dataset-2) are 0.984, 0.990, and 0.991 (0.977, 0.989, and 0.991) in the increasing order of the model size. All of the average overlaps are the same as 0.999, except the case of the smallest model with dataset-2, whose overlap is 0.998 (overlaps of 0.999 and 0.998 are equivalent to 0.1% and 0.2% errors). Namely, the model with 256 hidden neurons shows the highest accuracy.

TABLE A.1. Minimum and average overlap values of the test sub-dataset in dataset-1 and -2 according to models with the different number of hidden neurons.

The number of hidden neurons	64	128	256
Minimum overlap (dataset-1)	0.984	0.990	0.991
Minimum overlap (dataset-2)	0.977	0.989	0.991
Average overlap (dataset-1)	0.999	0.999	0.999
Average overlap (dataset-2)	0.998	0.999	0.999

Appendix B: Computing Time and Accuracy Variation of The Model According To $\mathcal{R}$

We examine how the number of elements $\mathcal{R}$ in an RNN cell affects the model in the aspects of computing time and accuracy. Table B.2 tabulates the typical elapsed time with a minimum overlap of each case on dataset-1 and -2. Although the model can speed up by increasing $\mathcal{R}$, the accuracy expense renders the model inapplicable for practical use.

TABLE B.2. Computation time and overlap variation with respect to the number of elements, $\mathcal{R}$, in a RNN cell.

$\mathcal{R}$	T_1	T_{1500}	Minimum overlap	
			dataset-1	dataset-2
1	$\mathcal{O}(10^{-1})$	$\mathcal{O}(1)$	0.991	0.991
10	$\mathcal{O}(10^{-2})$	$\mathcal{O}(10^{-1})$	0.913	0.910
100	$\mathcal{O}(10^{-3})$	$\mathcal{O}(10^{-2})$	0.823	0.805

REFERENCES

[1] B. P. Abbott *et al.* (Virgo, LIGO Scientific), Phys. Rev. Lett. **116**, 061102 (2016), arXiv:1602.03837 [gr-qc].

[2] J. Aasi *et al.* (LIGO Scientific), Class. Quant. Grav. **32**, 074001 (2015), arXiv:1411.4547 [gr-qc].

[3] F. Acernese *et al.* (VIRGO), Class. Quant. Grav. **32**, 024001 (2015), arXiv:1408.3978 [gr-qc].

[4] R. Abbott *et al.*, arXiv e-prints , arXiv:2010.14527 (2020), arXiv:2010.14527 [gr-qc].

[5] G. Turin, IRE Transactions on Information Theory **6**, 311 (1960).

[6] G. Lovelace *et al.*, Class. Quant. Grav. **33**, 244002 (2016), arXiv:1607.05377 [gr-qc].

[7] L. Blanchet, T. Damour, B. R. Iyer, C. M. Will, and A. Wiseman, Phys. Rev. Lett. **74**, 3515 (1995), arXiv:gr-qc/9501027 [gr-qc].

[8] S. Droz, D. J. Knapp, E. Poisson, and B. J. Owen, Phys. Rev. **D59**, 124016 (1999), arXiv:gr-qc/9901076 [gr-qc].

[9] A. Buonanno and T. Damour, Phys. Rev. **D62**, 064015 (2000), arXiv:gr-qc/0001013 [gr-qc].

[10] L. Blanchet, T. Damour, G. Esposito-Farese, and B. R. Iyer, Phys. Rev. Lett. **93**, 091101 (2004), arXiv:gr-qc/0406012 [gr-qc].

[11] M. Pürrer, Class. Quant. Grav. **31**, 195010 (2014), arXiv:1402.4146 [gr-qc].

[12] A. Taracchini *et al.*, Phys. Rev. **D89**, 061502 (2014), arXiv:1311.2544 [gr-qc].

[13] M. Pürrer, Phys. Rev. **D93**, 064041 (2016), arXiv:1512.02248 [gr-qc].

[14] A. Bohé *et al.*, arXiv preprint arXiv:1611.03703 (2016).

[15] P. Kumar, T. Chu, H. Fong, H. P. Pfeiffer, M. Boyle, D. A. Hemberger, L. E. Kidder, M. A. Scheel, and B. Szilagyi, Phys. Rev. D **93**, 104050 (2016).

[16] A. J. K. Chua, C. R. Galley, and M. Vallisneri, Phys. Rev. Lett. **122**, 211101 (2019).

[17] D. Williams, I. Siong Heng, J. Gair, J. A Clark, and B. Khamesra, "A precessing numerical relativity waveform surrogate model for binary black holes: A gaussian process regression approach," (2019).

[18] T. B. Brown *et al.*, arXiv e-prints , arXiv:2005.14165 (2020), arXiv:2005.14165 [cs.CL].

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, arXiv e-prints , arXiv:1706.03762 (2017), arXiv:1706.03762 [cs.CL].

[20] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, in *Proceedings of the 1st Annual Conference on Robot Learning* (2017) pp. 1–16.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Commun. ACM **60**, 8490 (2017).

[22] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, arXiv preprint arXiv:1406.1078 (2014).

[23] I. Sutskever, O. Vinyals, and Q. V. Le, in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14 (MIT Press, Cambridge, MA, USA, 2014) pp. 3104–3112.

[24] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, in *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (JMLR. org, 2017) pp. 1243–1252.

[25] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, in *Proceedings of the IEEE international conference on computer vision* (2015) pp. 4534–4542.

[26] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, arXiv preprint arXiv:1511.06114 (2015).

[27] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, *et al.*, arXiv preprint arXiv:1602.06023 (2016).

[28] A. Nitz *et al.*, "gwastro/pycbc: Pycbc v1.13.1 release," (2018).

[29] A. Bohé, L. Shao, A. Taracchini, A. Buonanno, S. Babak, I. W. Harry, I. Hinder, S. Ossokine, M. Pürrer, V. Raymond, T. Chu, H. Fong, P. Kumar, H. P. Pfeiffer, M. Boyle, D. A. Hemberger, L. E. Kidder, G. Lovelace, M. A. Scheel, and B. Szilágyi, Phys. Rev. D **95**, 044028 (2017).

[30] M. Boyle *et al.*, (2019), arXiv:1904.04831 [gr-qc].

[31] J. Healy, C. O. Lousto, J. Lange, R. O'Shaughnessy, Y. Zlochower, and M. Campanelli, arXiv preprint arXiv:1901.02553 (2019).

[32] K. Jani, J. Healy, J. A. Clark, L. London, P. Laguna, and D. Shoemaker, Classical and Quantum Gravity **33**, 204001 (2016).

[33] J. Healy, C. O. Lousto, Y. Zlochower, and M. Campanelli, Classical and Quantum Gravity **34**, 224001 (2017).

[34] A. Groné, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 1st ed. (O'Reilly Media, Inc., 2017).

[35] J. Sola and J. Sevilla, IEEE Transactions on Nuclear Science **44**, 1464 (1997).

[36] I. J. Tredcr, Astronomische Nachrichten **296**, 45 (1975), https://onlinelibrary.wiley.com/doi/pdf/10.1002/asna.19752960110.

[37] M. Favata, Phys. Rev. D **83**, 024028 (2011).

[38] Y. Bengio, R. Ducharme, and P. Vincent, in *Advances in Neural Information Processing Systems 13*, edited by T. K. Leen, T. G. Dietterich, and V. Tresp (MIT Press, 2001) pp. 932–938.

[39] S. Hochreiter and J. Schmidhuber, Neural computation **9**, 1735 (1997).

[40] F. A. Gers, J. Schmidhuber, and F. Cummins, (1999).

[41] R. Sturani, S. Fischetti, L. Cadonati, G. M. Guidi, J. Healy, D. Shoemaker, and A. Viceré, Journal of Physics: Conference Series **243**, 012007 (2010).

[42] W. Wei and E. A. Huerta, Physics Letters B **800**, 135081 (2020), arXiv:1901.00869 [gr-qc].

[43] M. Van Der Sluys, V. Raymond, I. Mandel, C. Röver, N. Christensen, V. Kalogera, R. Meyer, and A. Vecchio, Classical and Quantum Gravity **25**, 184011 (2008).

[44] J. Aasi *et al.* (LIGO-Virgo Scientific Collaboration), Phys. Rev. D **88**, 062001 (2013).

[45] S. A. Usman *et al.*, Class. Quant. Grav. **33**, 215004 (2016).

[46] B. P. Abbott *et al.* (LIGO Scientific, Virgo), Annalen Phys. **529**, 1600209 (2017), arXiv:1608.01940 [gr-qc].

[47] É. É. Flanagan and S. A. Hughes, Phys. Rev. D **57**, 4535 (1998), arXiv:gr-qc/9701039 [gr-qc].

[48] C. Cutler and M. Vallisneri, Phys. Rev. D **76**, 104018 (2007).

[49] The LSC-Virgo-KAGRA Observational Science Working Groups, *The LSC-Virgo-KAGRA Observational Science White Paper*, Tech. Rep. T2000424 (2020).