Generating higher order modes from binary black hole mergers with machine learning

Tim Grimbergen
Institute for Gravitational and Subatomic Physics (GRASP), Utrecht University, Princetonplein 1, 3584 CC Utrecht, The Netherlands

Stefano Schmidt,* Chinmay Kalaghatgi, and Chris van den Broeck
Institute for Gravitational and Subatomic Physics (GRASP), Utrecht University, Princetonplein 1, 3584 CC Utrecht, The Netherlands and Nikhef, Science Park 105, 1098 XG, Amsterdam, The Netherlands

(Received 12 February 2024; accepted 25 April 2024; published 20 May 2024)

We introduce a machine learning model designed to rapidly and accurately predict the time domain gravitational wave emission of nonprecessing binary black hole coalescences, incorporating the effects of higher order modes of the multipole expansion of the waveform. Expanding on our prior work [Phys. Rev. D **103**, 043020 (2021)], we decompose each mode by amplitude and phase and reduce dimensionality using principal component analysis. An ensemble of artificial neural networks is trained to learn the relationship between orbital parameters and the low-dimensional representation of each mode. Our model is trained with $\sim 10^5$ signals with mass ratio $q \in [1, 10]$ and dimensionless spins $\chi_i \in [-0.9, 0.9]$, generated with the state-of-the-art approximant SEOBNRV4HM, and it is able to generate waveforms up to $\sim 4 \times 10^5 M$ long. We find that it achieves a median faithfulness of 10^{-4} averaged across the parameter space. We show that our model generates a single waveform 2 orders of magnitude faster than the training model, with the speedup increasing when waveforms are generated in batches. This framework is entirely general and can be applied to any other time domain approximant capable of generating waveforms from aligned spin circular binaries, possibly incorporating higher order modes.

DOI: 10.1103/PhysRevD.109.104065

I. INTRODUCTION

With almost a hundred of confirmed detections, gravitational wave (GW) astronomy is entering a mature state, where many loud GW events will force the scientific community to develop faster analyses to deliver precision measurements. Expanding on past results [1–3], the recent transient catalog GWTC-3 [4] is the latest achievement of the effort carried on by the LIGO-Virgo-KAGRA collaboration [5–8] and it relies on both instrument and data analysis development.

A crucial element of the data analysis is the ability to quickly and accurately generate waveforms for GW signals emitted by coalescing binary black holes (BBHs). Such waveforms are used for the expensive Bayesian estimation of the parameters characterizing a BBH [9]: the analysis of a single event requires the online generation of up to billions of waveforms. As we move towards the next generation of detectors, such as Einstein Telescope [10,11] and Cosmic Explorer [12,13], it will become paramount to deploy accurate waveform models that are fast and, at the same time, incorporate the full physics of the problem, otherwise our analyses will become subject to systematic errors in the parameter recovery [14]. On the other hand, accurate models are often slow to generate on a computer and the analyses might struggle to keep up with the large event rate expected in the next-generation observatories [15]. Balancing the two needs is challenging, since speed and accuracy are often at trade.

An essential aspect for a realistic BBH signal model is the incorporation of higher-order modes (HMs) of the multipole expansion of the waveform [16]. For nearly equal mass systems, the leading-order mode is orders of magnitude larger than the others and, including the HMs, does not significantly affect the parameter estimation. However, it has been demonstrated [17–20] that HMs are observable in highly asymmetric binary systems. In fact, the effect of HMs has already been observed in at least two BBH events originating from asymmetric binaries [21,22]. This underscores the importance of including HMs in any parameter estimation pipeline in order to avoid biases in the recovered parameters.

Two main families of models have been developed, both being able to incorporate HMs. One family relies on the

*s.schmidt@uu.nl

effective one body (EOB) formalism [23–30], which maps the complicated general relativistic binary system into a problem governed by an effective Hamiltonian. EOB models tend to be accurate but are quite costly to generate, since for each waveform one needs to solve the Hamiltonian equation of motion. On the other hand, the phenomenological waveforms [31–33] are based on analytical expressions, making use of the post-Newtonian formalism to model the inspiral, and on fits to numerical simulations to describe the intermediate and merger-ringdown regimes. They tend to be faster to evaluate than the EOB models. Both families, EOB and phenomenological, need to be calibrated with numerical relativity waveforms, computed by directly solving the Einstein equations in discretized form. The calibration makes sure that a model retains its accuracy even close to merger, where approximate treatments such as the post-Newtonian or EOB formalisms are no longer applicable by themselves.

Besides the standard families, surrogate waveform models have been developed with the aim of reproducing the output of a target model and of making feasible the usage of the underlying model. A first class of surrogates is designed to closely reproduce numerical relativity (NR) waveforms [34–40] and, accordingly, it is trained using only NR waveforms as input. NR surrogates are very accurate but they tend to be very short, due to the nature of the NR waveforms employed for training. For this reason, they are often hybridized using an analytical expression for the early inspiral. Besides targeting NR waveforms, a second class of surrogates has been developed to accelerate EOB models [41–45], even including HMs. While traditional surrogate models build an empirical interpolant on the waveform space, a more recent approach relies on performing a regression using machine learning techniques [46–48].

Among others, [49] introduced a machine learning surrogate model, based on a dimensionality reduction scheme followed by a regression. The framework was later applied to the generation of frequency domain signals from binary neutron star (BNS) systems [50]. In this work, we extend the model to HMs and we improve the accuracy of the regression by employing artificial neural networks (ANN). Our model marks a step towards the development of a faster, yet precise, waveform model, and will help enable the accurate analysis of next-generation detector data.

We train our model on the widely used approximant SEOBNRV4HM [25] to target systems with mass ratio $q \in [1, 10]$ and dimensionless spin components between $[-0.9, 0.9]$. Our model is able to generate waveforms with a maximum length of $t = 2\frac{M}{M_\odot}s$, which amounts to $t \simeq 4.06 \times 10^5 M$ in geometrized units, and achieves 10^{-4} median faithfulness (with tails up to 10^{-2}) when averaged across a wide range in parameter space. Our numerical experiments show that our model offers a substantial speedup with respect to the original model, matching the speed of the state-of-the-art surrogate models.

Our methodology distinguishes itself from previous approaches [46–48] in several ways. First of all, to predict the phase $\phi_{\ell m}$ of each mode, we use three separate models for predicting the basis coefficients of the principal component decomposition. This "distribution of tasks" allows for more flexibility, a significant reduction in the total number of model parameters, and fewer training waveforms. Second, we improve on choosing the hyperparameters of the ANNs. Whereas previous approaches arrive at their configuration of hyperparameters using somewhat limited heuristics, we introduce a more rigorous method by using Bayesian optimization to tune hyperparameters. Lastly, and more importantly, even though we have not incorporated precession, we show that our approach is viable for the large range of parameters $q \times \chi_{1z} \times \chi_{2z} = [1, 10] \times [-0.9, 0.9] \times [-0.9, 0.9]$. Previous approaches either focused solely on the dominant mode of nonprecessing spin-aligned waveforms [46,47] or on precessing HM waveforms but with a limited range in mass ratio $q \in [1, 2]$ and with $\chi_2 = 0$ [48].

This paper is organized as follows. In Sec. II we introduce the details of the model presented here, stressing the differences with the model in [49]. Section III is devoted to the validation of our model: we will motivate our choice of several hyperparameters and perform an accuracy and speed study. In Sec. IV, we present some final remarks and highlight future perspectives.

II. BUILDING THE MODEL

A nonprecessing BBH can be described by four *intrinsic* parameters, which specify the two BH masses m_1 and m_2 and the z components of the two dimensionless spins, χ_{1z} and χ_{2z}. Since the total mass $M = m_1 + m_2$ acts as a scaling parameter, when generating nonprecessing BBH signals, one only needs to consider the mass ratio $q = m_1/m_2 \geq 1$ together with the spins. We refer to the relevant parameters as $\boldsymbol{\vartheta} = (q, \chi_{1z}, \chi_{2z})$. Besides the masses and spins, the gravitational wave emitted by the system depends also on luminosity distance to the source d_L, the inclination angle ι of the source, and the reference phase φ_0; these are the *extrinsic* parameters.

As is standard, we expand the angular dependence on ι, φ_0 of the *complex* waveform $h(t)$ in terms of a sum of spin-2 spherical harmonics. A GW is then parametrized[1] as [51]

$$\begin{aligned} h(t; d_L, \iota, \varphi_0, m_1, m_2, \chi_{1z}, \chi_{2z}) &= h_+ + i h_\times \\ &= \frac{G}{c^2}\frac{M}{d_L}\sum_{\ell=2}^{\infty}\sum_{m=-\ell}^{\ell} {}^{-2}Y_{\ell m}(\iota, \varphi_0) h_{\ell m}(t/M; \boldsymbol{\vartheta}), \end{aligned} \quad (1)$$

[1]Such parametrization is particularly convenient as it separates the waveform dependence over intrinsic and extrinsic parameters.

where we refer to the functions $h_{\ell m}(t;\boldsymbol{\vartheta})$ as *modes* of the waveform. We note that, for nonprecessing systems, $h_{\ell m} = (-1)^{\ell} h^*_{\ell -m}$, hence we will only consider modes with $m > 0$.

The mode $(\ell, m) = (2, 2)$ is the largest in amplitude, hence it is often referred to as the *dominant mode*. The other subdominant modes are usually a few orders of magnitude smaller in amplitude and become more relevant (and measurable) for high mass ratios [20–22].

In this work, we introduce a machine learning model to perform a regression,

$$(q, \chi_{1z}, \chi_{2z}) \longmapsto h_{\ell m}(t;\boldsymbol{\vartheta}), \tag{2}$$

for each mode (ℓ, m). The regression is designed to reproduce waveforms from a given dataset; such waveforms can be generated by *any* time-domain approximant.

We decompose each mode in an amplitude term $A_{\ell m}$ and a phase term $\phi_{\ell m}$ as follows:

$$h_{\ell m}(t;\boldsymbol{\vartheta}) = A_{\ell m}(t;\boldsymbol{\vartheta}) e^{i\phi_{\ell m}(t;\boldsymbol{\vartheta})}, \tag{3}$$

and, for each mode, we perform a regression for amplitude and phase separately. The regression scheme closely follows [49] and relies on:

(a) a suitable vector representation of the regression target by choosing a fixed time grid;
(b) a principal component analysis (PCA) model to reduce the dimensionality of each waveform;
(c) an artificial neural network (ANN) regression to learn the dependence on $\boldsymbol{\vartheta}$ of the reduced waveform.

While the first two elements are unchanged from the previous work, the ANN regression is first introduced here. Indeed a NN has more representation power than the mixture of experts model [52], used in [49]: the change was needed to achieve better accuracy for the model.

A. Dataset creation

To construct a dataset, we follow [49] and we set a dimensionless time grid. We construct the grid by setting D points equally spaced in τ^{α}, where τ is the physical time scaled by the total mass of the system M: $\tau = t/M$. Using the findings of [49], we set $D = 2000$ and $\alpha = 0.5$. This is a good compromise between the need of having a faithful representation of the waveform (which requires a large grid) and the need of having a compact model (which points to a sparse grid). The waveforms are time shifted so that the peak of the amplitude of the (2, 2) mode happens at $\tau = 0$. The grid starts at (scaled) time $\tau_{\min} = -\tau_0$, where τ_0 sets the length of the waveform that our model is able to generate (as a function of the total mass M). We choose $\tau_0 = 2s/M_{\odot}$, which in geometrized units (i.e., with $G = c = 1$) amounts to a waveform length of $t \simeq 4.06 \times 10^5 M$. We populate the dataset with 68000 waveforms.

To make sure that the distribution of q is skewed towards the boundaries, where the regression is less accurate, we sample the mass ratio q in the range [1, 10] with the following procedure:

(i) We sample $q_1, \ldots, q_5 \sim \mathcal{U}_{[1,10]}$.
(ii) We sample $x \sim \mathcal{U}_{[0,1]}$.
(iii) We select q, based on the value of x,
if $x \in [0, 0.3)$, $\min q_1, \ldots, q_5$,
if $x \in [0.3, 0.8)$, q_1,
if $x \in [0.8, 1]$, $\max q_1, \ldots, q_5$,

where $\mathcal{U}_{[a,b]}$ is the uniform distribution in $[a, b]$. The spins are drawn uniformly in the range $[-0.9, 0.9]$.

Once a time grid is set, we evaluate all the modes (amplitude and phase) on the time grid and represent them as vectors in $\mathbb{R}^D$. We then create a dataset $\{X, Y\}$ of N elements. Each row of the dataset is of the form

$$X = [q, \chi_{1z}, \chi_{2z}] \tag{4}$$

$$Y = [\boldsymbol{A}^T_{\ell m}, \boldsymbol{\phi}^T_{\ell m}, \ldots]. \tag{5}$$

The dataset Y gathers the amplitude and phase for the different modes in the dataset. We include all the modes available in SEOBNRV4HM: $(\ell, m) = \{(2,2), (2,1), (3,3), (4,4), (5,5)\}$. In what follows we will refer to any of the vectors $\boldsymbol{A}_{\ell m}$ or $\boldsymbol{\phi}_{\ell m}$ as $\boldsymbol{f}$. Note that we use the same grid for all the modes.

It is well known [34,35] that in the case of a symmetric BBH, where $m_1 \simeq m_2$ and $\chi_{1z} \simeq \chi_{2z}$, the amplitude of the odd-m modes vanishes, making the phase Eq. (3) ill defined. Clearly this is a challenge for the regression, which is likely to perform poorly in such situations. While several alternatives are available in the literature, we will address the challenge in future work. For the moment we will content ourself with a poor performance of the regression for the odd-m modes in the $m_1 \simeq m_2$ region, as we will show in Figs. 2 and 4. This will have little impact for the overall faithfulness, due to the small amplitude of such modes.

B. Dimensionality reduction

It is unfeasible to perform a regression targeting a high-dimensional vector such as $\boldsymbol{f} \in \mathbb{R}^D$. For this reason, in [49] we introduced a principal component analysis (PCA) dimensionality reduction scheme. It is an *approximately* invertible linear mapping between a vector $\boldsymbol{f} \in \mathbb{R}^D$ in a large dimensional space to lower dimensional vector $\boldsymbol{g} \in \mathbb{R}^K$:

$$\mathbf{g} = H(\mathbf{f} - \boldsymbol{\mu}), \tag{6}$$

$$\hat{\mathbf{f}} = H^T \mathbf{g} + \boldsymbol{\mu}, \tag{7}$$

where $\boldsymbol{\mu} \in \mathbb{R}^D$ and H is a $K \times D$ matrix. The rows $H_{i:}$ of H, also called *principal components* (PC), form an orthonormal set of vectors, i.e., $\sum_{k=1}^{D} H_{ik}H_{kj} = \delta_{ij}$. The PCs are the first K eigenvectors of the $D \times D$ covariance matrix of the dataset, as described in Sec. 12 of [53].

The mapping is only approximately invertible, in the sense that $\hat{\mathbf{f}}$ is only an approximation of the high dimensional vector $\mathbf{f}$. The quality of the approximation is controlled by the number K of PCs considered: the more PCs, the more accurate the reconstruction of $\mathbf{f}$ is.

One can have a deeper insight on PCA considering the following formula for the reconstructed vector $\hat{\mathbf{f}}$ (setting $\boldsymbol{\mu} = 0$ without loss of generality):

$$\hat{\mathbf{f}} = \sum_{i=0}^{K-1} \langle \mathbf{f} | H_{i:} \rangle H_{i:}, \tag{8}$$

where $\langle \mathbf{a} | \mathbf{b} \rangle = \sum_{i=0}^{D-1} a_i b_i$ is the *Euclidean* scalar product between two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^D$. Since less important PCs are more orthogonal to data, the typical magnitude of $g_i = \langle \mathbf{f} | H_{i:} \rangle$ decreases as i increases.[2] As a consequence, the regression for a lower order PC needs to be more accurate than the one for the higher order PC. This will be taken care of by a suitable choice for the loss function for the regression (see next section).

Following [49], in this work we employ for each mode six PCA components for the phase model and four for the amplitude. While it is plausible that an optimal number of components may vary for different modes, we opt for simplicity by employing the same number of PCA components for all modes, a configuration tuned based on the (2, 2) mode only.

C. Neural network regression

An artificial neural network (ANN) is a popular regression model, consisting of a powerful parametric function, whose parameters (or weights), when properly set, can represent a large variety of relations between input and output [53–55]. An ANN is built by stacking together N_L layers in such a way that the output of a layer is the input of the following layer. Each layer is a function $L\colon \mathbb{R}^{D'} \to \mathbb{R}^{D''}$ and has the following functional form:

$$\boldsymbol{y} = a(W'_{\prime\prime}\boldsymbol{x}), \tag{9}$$

where $W'_{\prime\prime}$ is a $D'' \times D'$ matrix and $a\colon \mathbb{R} \to \mathbb{R}$ is an activation function that acts elementwise on the vector $W'_{\prime\prime}\boldsymbol{x}$. Each component y_i of the output of the layer is called a node and the number of nodes is a tunable parameter, controlling the representative power of the layer.

[2]For this reason PCA can be seen as a perturbative expansion on the basis vectors $H_{i:}$, where the accuracy is roughly measured by the eigenvalues of the first neglected PC. Increasing the number K of PCs considered increases the accuracy of the model (but also the complexity of the model).

An ANN $\mathcal{N}$ is obtained by composing N_L different layers (each with a suitable number of nodes):

$$\mathcal{N}_W = L_{N_\mathrm{L}} \circ \cdots \circ L_2 \circ L_1, \tag{10}$$

where we denote by W the set of all the parameters the ANN depends on.

The number of layers, together with the number of nodes per layer, are hyperparameters that need to be carefully chosen, to balance model accuracy and model complexity. Another important choice is the activation function: several possible choices are possible, the most popular being the *sigmoid*, the hyperbolic tangent or the so-called ReLU function. In our work, we consider the sigmoid function between all layers, except for the very last layer which has linear/identity activation so that negative values are also possible.

Once the ANN is set up, we need to set its weights to the values that achieve our regression task. This procedure is called training, where we minimize a loss function with respect to the weights $\boldsymbol{W}$ of the model. The loss function depends on the dataset at hand $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}$. Mathematically, the weights are given by

$$\boldsymbol{W} = \underset{W}{\operatorname{argmin}}\, \mathcal{L}(W; \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_i). \tag{11}$$

The minimization of the loss function is performed by stochastic gradient descent, as implemented by the NADAM algorithm [56], which combines the popular ADAM algorithm [57] with the Nesterov momentum. The optimization relies on the gradients $\partial_W \mathcal{L}$ of the loss function, computed through the back-propagation algorithm [58].

To perform our regression $\theta \mapsto \boldsymbol{g}$, we employ an ensemble of networks that suitably combined delivers accurate results. To improve the representative power or the ANN, we employ feature augmentation on the vector $\boldsymbol{\vartheta} = (q, \chi_{1z}, \chi_{2z})$, effectively using the augmented vector $\tilde{\boldsymbol{\vartheta}}$ as input for the regression. Although different ANNs will need different features, we will for convenience abuse the notation $\tilde{\boldsymbol{\vartheta}}$ to denote any augmented vector. Indeed, the features to add need to be chosen with a validation process: this will be discussed in the next section.

Before the training, the regression targets $\boldsymbol{y}_i$ are scaled such that $\boldsymbol{y}_i \to \frac{\boldsymbol{y}_i}{\boldsymbol{w}}$, where $\boldsymbol{w}$ keeps the maximum of $|\boldsymbol{y}_i|$ along each axis. In this way all the regression targets span the same order or magnitude, facilitating the "learning" task.

For the amplitude $\boldsymbol{A}_{\ell m}$ of each mode, we employ a single ANN $\mathcal{N}_{A_{\ell m}}$ that predicts the first four PCA components. The predicted amplitude $\hat{\boldsymbol{A}}_{\ell m}$, including the PCA reconstruction, has the following form:

$$\hat{\boldsymbol{A}}_{\ell m}(\boldsymbol{\vartheta}) = \boldsymbol{\mu}_{A_{\ell m}} + H^T_{A_{\ell m}} \mathcal{N}_{A_{\ell m}}(\tilde{\boldsymbol{\vartheta}}). \tag{12}$$

For the phase $\boldsymbol{\phi}_{\ell m}$, we employ one ANN $\mathcal{N}_{\phi_{\ell m}\text{-}01}$ to predict only the first two PCA components. Another ANN will take care of the remaining components $\mathcal{N}_{\phi_{\ell m}\text{-}2345}$. On top of this, we build an additional ANN $\mathcal{N}_{\phi_{\ell m}\text{-residual}}$ to target the residual of the predictions of $\mathcal{N}_{\phi_{\ell m}\text{-}01}$. The scheme makes sure that the first two PCs are predicted with much larger accuracy than the others. Indeed, the reconstructed waveform depends largely on the first two components and a small fractional error can potentially have a large impact on the overall accuracy.

The predicted phase $\hat{\boldsymbol{\phi}}_{\ell m}$ is then given by

$$\hat{\boldsymbol{\phi}}_{\ell m}(\boldsymbol{\vartheta}) = \boldsymbol{\mu}_{\phi_{\ell m}} + H^T_{\phi_{\ell m}} \begin{pmatrix} \mathcal{N}_{\phi_{\ell m}\text{-}01}(\tilde{\boldsymbol{\vartheta}}) + \mathcal{N}_{\phi_{\ell m}\text{-residual}}(\tilde{\boldsymbol{\vartheta}}) \\ \mathcal{N}_{\phi_{\ell m}\text{-}2345}(\tilde{\boldsymbol{\vartheta}}) \end{pmatrix}. \tag{13}$$

We train our model using the PCA dataset, obtained by PCA reducing the training set. Each ANN is trained using the following loss function:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}((\mathcal{N}(\boldsymbol{\vartheta}_i) - \boldsymbol{y}_i))^2 \boldsymbol{w}, \tag{14}$$

where $\boldsymbol{y}_i$ is the (scaled) regression target of each network and $\boldsymbol{w} \in \mathbb{R}^K$ takes into account the fact that different PCs have different orders of magnitude.

The network is implemented and trained using the python package KERAS [59], built on TensorFlow backend [60].

III. PERFORMANCE STUDY

In this section, we first study how the model performance depends on the different choices of hyperparameters (network architecture, learning rate, features, …). The architecture details of the model (chosen after hyperparameters tuning) are reported in Table I. We then evaluate the faithfulness of our model and report the speedup that we obtain when using our surrogate instead of the training model SEOBNRV4HM. In what follows, we will refer to our model as MLGW-SEOBNRV4HM.

TABLE I. Architecture of the four ANNs employed to generate each mode. For each ANN we report the number of layers and the number of units per layer. We perform data augmentation by adding all the polynomial terms in the chosen features. The architecture has been chosen after hyperparameter tuning (see Fig. 1). Among other features, we use the chirp mass $\mathcal{M}_c = \frac{(m_1 m_2)^{3/5}}{(m_1+m_2)^{1/5}}$, the symmetric mass ratio $\eta = \frac{m_1 m_2}{(m_1+m_2)^2}$, and the effective spin parameter $\chi_{\rm eff} = \frac{m_1\chi_{1z}+m_2\chi_{2z}}{m_1+m_2}$.

Network	n-layers	Units	Features	Order
$\mathcal{N}_{A_{\ell m}}$	1	35	$\mathcal{M}_c, \chi_{\rm eff}$	1
$\mathcal{N}_{\phi_{\ell m}\text{-}01}$	2	50	$\mathcal{M}_c, \eta, \log q, \chi_{\rm eff}$	3
$\mathcal{N}_{\phi_{\ell m}\text{-}2345}$	1	50	$\mathcal{M}_c, \eta, \log q, \chi_{\rm eff}$	1
$\mathcal{N}_{\phi_{\ell m}\text{-residual}}$	5	50	$\mathcal{M}_c, \eta, \log q, \chi_{\rm eff}$	2

To measure the discrepancy between two waveforms h_1, h_2, we define a scalar product,

$$(h_1|h_2) = 4\Re\int_0^\infty \mathrm{d}f\,\frac{\tilde{h}^*{}_1(f)\tilde{h}_2(f)}{S_n(f)}, \tag{15}$$

where $\tilde{}$ denotes the Fourier transform and $S_n(f)$ is the power spectral density (PSD) of the detector's noise. We can use the scalar product to arrive at a normalized waveform, $\hat{h} = \frac{h}{\sqrt{(h|h)}}$.

To measure the discrepancy between two individual modes $h^1_{\ell m}$ and $h^2_{\ell m}$, we define the *match* $\mathcal{M}$:

$$\mathcal{M} = \max_{t,\phi}(\hat{h}^1_{\ell m}|\hat{h}^2_{\ell m}e^{i2\pi f t + i\varphi}), \tag{16}$$

where $he^{i2\pi f t+i\varphi}$ denotes (with a slight abuse of notation) h translated in time by a factor of t and with its phase shifted by φ. We call *mismatch* the quantity $\mathcal{F} = 1 - \mathcal{M}$.

The match defined above amounts to the search statistics being used for matched filtering searches of nonprecessing/non-HM signals [61]. A different statistic is needed to search for HM signals, hence the match defined above is not suitable to compare two different waveforms with HM content as in Eq. (1). In this case, we need to compare the two polarizations h_+, $h_\times$ of a waveform with a signal s observed at the detector:

$$s = F_+ h_+ + F_\times h_\times, \tag{17}$$

where F_+, $F_\times$ are called antenna pattern functions, depending on the sky location of the source and on the polarization angle [16].

Following [62], we introduce the *symphony match* between a signal s and a waveform h:

$$\mathcal{M}_{\rm sym} = \max_t \frac{(\hat{s}|\hat{h}_+)^2 + (\hat{s}|\hat{h}_\times)^2 - 2(\hat{h}_\times|\hat{h}_+)(\hat{s}|\hat{h}_+)(\hat{s}|\hat{h}_\times)}{1-(\hat{h}_\times|\hat{h}_+)^2}. \tag{18}$$

Note that $\mathcal{M}_{\rm sym}$ depends on the signal s, hence it depends on the sky location and polarization angle. As above, we define the symphony mismatch as $\mathcal{F}_{\rm sym} = 1 - \mathcal{M}_{\rm sym}$.

In what follows, we always use a constant (i.e., flat) PSD. While this certainly does not correspond to any actual detector, it makes sure that all the frequencies are weighted equally, hence giving a detector agnostic measure of the mismatch.

A. Hyperparameter tuning

The performance of the model depends on a number of crucial choices about some nontrainable parameters, usually called hyperparameters. The hyperparameters define the architecture of the ANN as well as some parameters relevant to the training. Setting the right values for the hyperparameters is crucial for the ANN performance, as one needs to balance between accuracy and speed; this procedure is called *hyperparameter tuning* and can be done automatically to optimize manual work and to make sure to find a good minimum.

We optimize the following hyperparameters:

(i) n-layers: number of hidden layers in the ANN;
(ii) units: number of nodes per hidden layer;
(iii) features: features to use for data augmentation;
(iv) order: the data will be augmented with all the monomials of the chosen features up the given order.

For each of the four ANNs useful to produce a single mode [see Eqs. (12) and (13)], we train a network for different combinations of hyperparameters. The figure of merit of each hyperparameter choice is the logarithm of the loss function [see Eq. (14)] evaluated on the validation set. For our experiments we only use the dataset of the (2, 2) mode and we employ the package KERAS-TUNER [63]. Specifically, we use the Bayesian optimization tuner which somewhat prioritizes searching around more promising configurations.

We report our results in Fig. 1, where each combination of hyperparameters tested is represented in the n-layers–units plane and colored by the validation score. We can see that all four ANNs share the same trend: the most effective way to improve regression accuracy is to increase the number of units as opposed to the number of layers. The number of layers is far more important than extra features or the polynomial order for data augmentation.

Furthermore, we note that the regressions for the amplitude and for the high phase PCs (i.e., components 2–5) can be performed with a smaller model, compared to the models for the first two PCs of the phase. This can be explained by the fact that most of the physical information is stored in the first two components of the phase, making this a harder regression problem.

In Table I we report the final hyperparameter choice we made for each of the networks. The architectures are the same across the different modes considered.

As discussed above, we note that models $\mathcal{N}_{A_{\ell m}}$ and $\mathcal{N}_{\phi_{\ell m}\text{-}2345}$ are very simple, having only one layer and a small polynomial order, while the other ANNs have a more complicated architecture. We note here that an accurate ANN for the residuals of the phase is crucial to obtain a good accuracy: indeed $\mathcal{N}_{\phi_{\ell m}\text{-residual}}$ is the most complex model we employ, meaning that the residual phase dataset is the “hardest” to learn.

B. Accuracy study

To test the accuracy of our model, we generate a test set with 50000 randomly chosen waveforms generated with the training model SEOBNRV4HM. The waveform masses are characterized by a total mass $M = 20M_\odot$ and by a mass ratio $q \in [1, 10]$, while the spins are chosen in the range $[-0.9, 0.9]$. The inclination angle ι and reference phase φ_0 are drawn uniformly from a sphere. To vary the length in time of the waveforms considered, we sample the starting frequency $f_{\min}$ uniformly in the range [15, 75] Hz.

In Fig. 2, we report the histogram of the distribution of the mismatches between MLGW-SEOBNRV4HM and the test waveforms. The upper part refers to the mismatches Eq. (18) computed on the overall waveforms (with sky location sampled uniformly over the sky); the lower box refers to mismatches computed mode by mode with Eq. (16).

First of all, we note that the model shows very high faithfulness. With a median value of 4×10^{-4} and with virtually no signals with a symphony mismatch higher than 10^{-2}, the accuracy of MLGW-SEOBNRV4HM matches the accuracy of other state-of-the-art surrogate models [44,45,47] and the accuracy of the training model SEOBNRV4HM in reproducing numerical relativity waveforms [25]. The faithfulness for the (2, 2) mode is even higher, with no signals with mismatch exceeding 2×10^{-3}. On the other hand, the higher order modes are less accurately reproduced than the dominant mode. In particular, for the modes (2, 1), (3, 3), (5, 5) a limited number of waveforms show very high mismatches $\mathcal{O}(1)$. See below for more discussion.

In Fig. 3 we report the dependence of the symphony mismatch as a function of the different orbital parameters. From the figure, it is manifest that the model has very stable performance across the parameter space. The faithfulness decreases for high positive values of the spin of the first object s_{1z} and for mass ratio $q \sim 1$. Despite this, in such “extreme” regions, the average mismatch is still of the order of 10^{-4}. The performance of the regression does not depend on χ_{2z}, since the quantity plays a very little role in defining the waveform features.

Longer waveforms, characterized by a lower $f_{\min}$, tend to show higher faithfulness. As longer waveforms are dominated by the inspiral phase, we can conclude that our model is more successful in reproducing the inspiral as opposed to the merger and the ringdown, prevalent in short waveforms. This feature is very important for the extension of our model to longer time grids, a necessity for analyzing binary neutron star systems or for applications in next-generation detectors [14].

In Fig. 4, for each mode we report the mismatch as a function of the mass ratio and of s_{1z}. One more time, we can see that the model faithfulness decreases for low mass ratios and for high spins. Moreover, the subdominant modes show a poorer performance as compared to the dominant one: this was already observed in Fig. 2.

The observed decrease in faithfulness for subdominant modes needs some attention. As discussed in Sec. II A, symmetric systems with $q \simeq 1$ have a vanishing amplitude of the odd-m modes and a poorly defined phase and, as

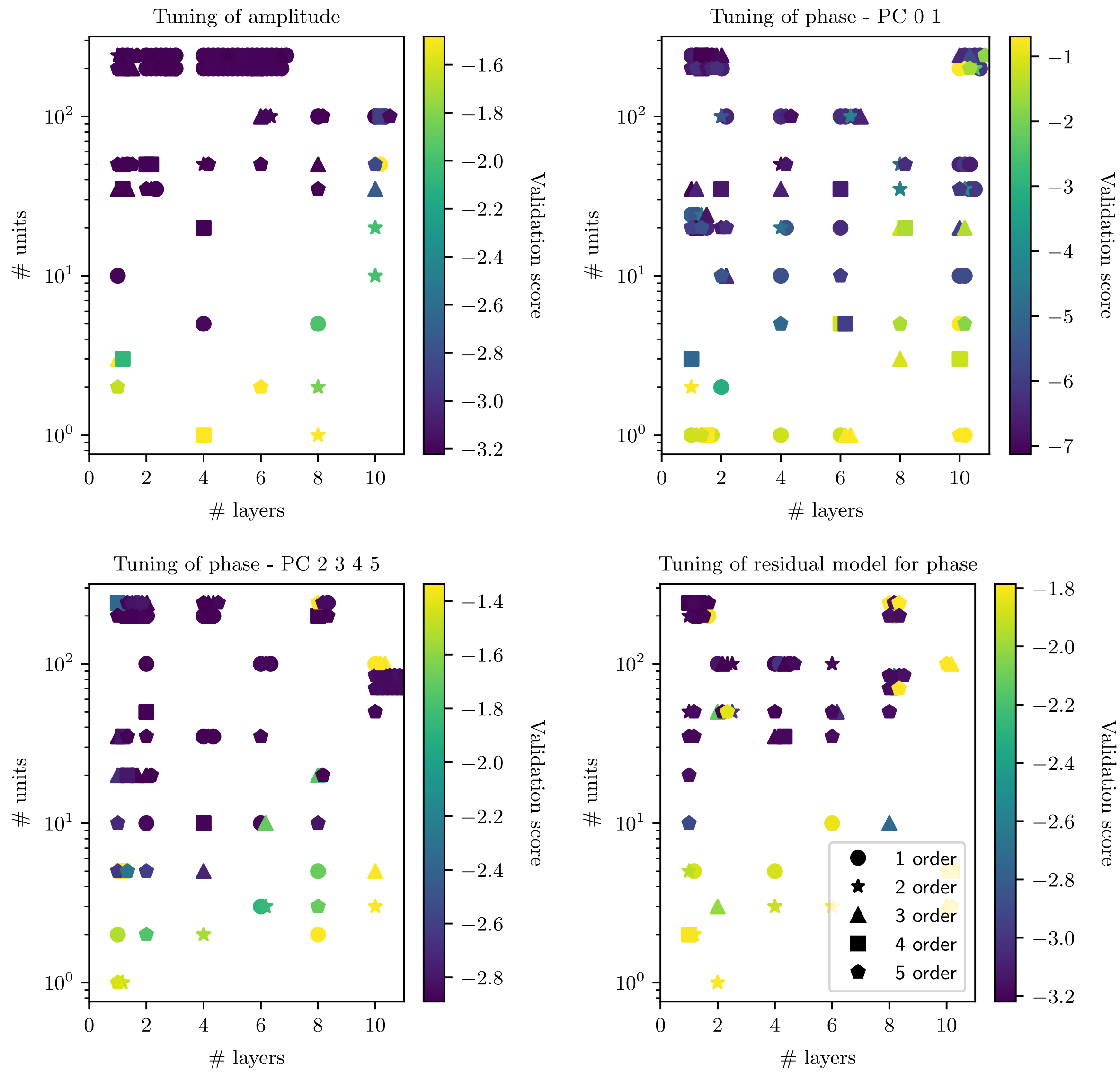

FIG. 1. Results from the validation of our ANN models, using the $\ell, m = 2, 2$ mode dataset. We tune the number of layers and the number of features per layer, together with the features and the polynomial order for the data augmentation. Each panel in the figure refers to a different ANN, taking care of different parts of the regression, as described in Sec. II C. For each regression, we train 100 ANNs with different choices of hyperparameters. Each point in the plot refers to a trained network and it is colored with the logarithm of the loss function computed on the validation data, referred to as the *validation score*. Note that we do not report the features used for data augmentation, so that the plot is degenerate in this quantity.

such, they correspond to "outliers" in the dataset. This clearly poses a challenge for both the PCA and the regression model for the amplitude, since modeling such a sharp feature of the data requires an enhanced model flexibility and more training examples. This is consistent with the low performance of the fit at low q observed in Fig. 4.

This matter is well known and several mitigation strategies are available in the literature. First of all, we might incorporate the vanishing behavior of the amplitude in the functional model for the regression, as done in [34]. Concretely, we could introduce a q-dependent amplitude scaling for the waveforms before adding them to the dataset, resulting in a dataset with amplitude time series of approximately the same magnitude. Second, we might mitigate the effect of a poorly defined phase by transforming all the modes except the (2,2) in the co-orbital frame [35] according to

$$h_{\ell m} \to h_{\ell m} e^{-i\frac{m}{2}\phi_{22}}, \tag{19}$$

where ϕ_{22} is the phase of the (2, 2) mode. Both of the strategies above reduce the outlier nature of waveforms with $q \sim 1$ and will likely improve the quality of the fit.

Another straightforward alternative could deploy a larger network for such modes. Indeed, we tuned the

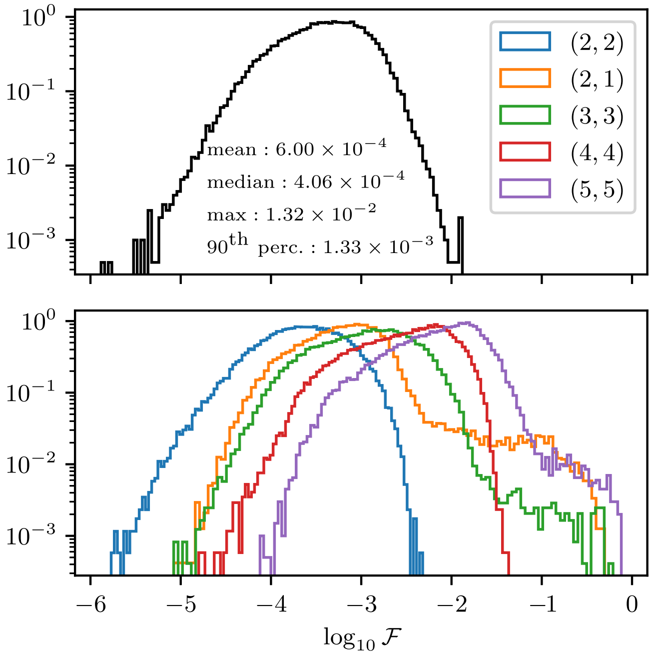

FIG. 2. We report the results of the mismatch between the 50000 test waveforms produced by MLGW-SEOBNRV4HM and by the training model SEOBNRV4HM. In the top panel, we report the histogram of the "symphony" mismatch F_{sym} for the overall waveforms, where we compare the h_+ and $h_\times$ polarizations [see Eq. (8)]. For the computation, we set random sky location. We also report the median, the mean, and the maximum mismatch, together with the value of the 90th percentile. In the bottom panel, we report the histograms for the mismatches computed mode by mode. The composition of the test set is described in the text.

hyperparameters on the (2, 2) mode (an "easy" regression target). Performing a network tuning on the dataset of HMs might reveal that our chosen architecture is not optimal.

Finally, we note that since for $q \sim 1$, the subdominant modes have a vanishing amplitude, a large mismatch in the subdominant mode for $q \sim 1$ has very little impact on the overall waveform Eq. (1), as shown in Fig. 2. This explains why the overall mismatch is low, despite high mismatch for the HMs in some edge cases.

C. Timing study

A speedup in the waveform generation is the main motivation to build a machine learning waveform generator; for this reason it is crucial to assess the gain in waveform generation time. For this reason, we use our test set to measure the ratio between the time to generate a waveform with SEOBNRV4HM and MLGW-SEOBNRV4HM. Our model offers further speed by generating waveform in batches: in this case, some operations are efficiently parallelized and happen more efficiently. We report our findings in Fig. 5.

We achieve a speedup ranging between a factor of 150 and 250, depending on the waveform characteristics. When waveforms are generated in batches of 100, the speedup can be substantially larger, reaching up to 1200, although with considerable variance, mostly due to the waveform length.

The speedup achieved by MLGW-SEOBNRV4HM is comparable to the one obtained by the SEOBNRV4HM_ROM surrogate model [44], which is the state-of-the-art frequency domain surrogate model trained on SEOBNRV4HM. SEOBNRV4HM_ROM is obtained with standard techniques and it achieves a speedup ranging between 100 and 200. The two results might not be directly comparable, since the

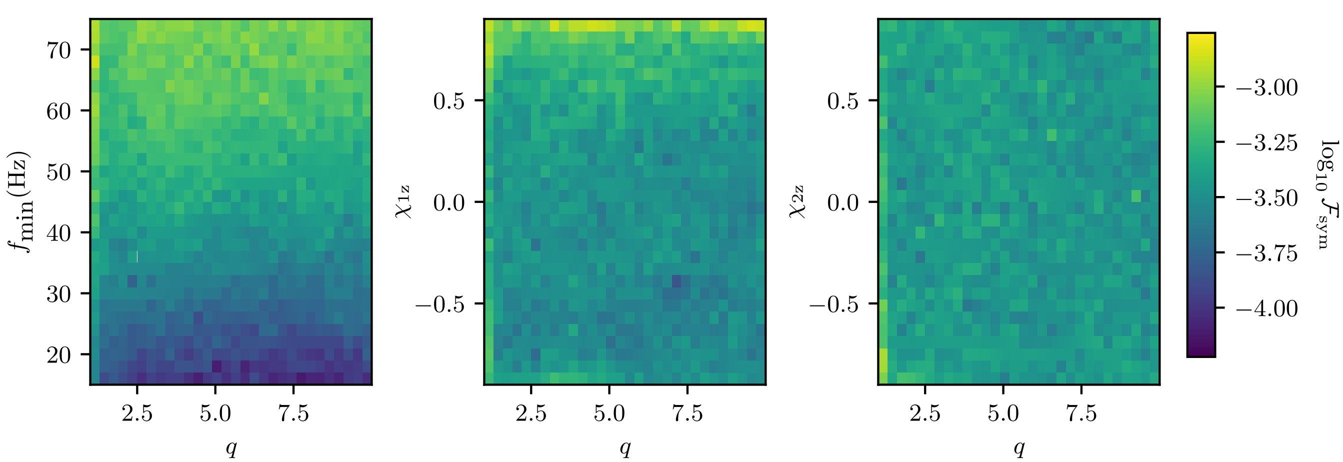

FIG. 3. Dependency of the symphony mismatch F_{sym} between MLGW-SEOBNRV4HM and the training model SEOBNRV4HM, as a function of some chosen waveform orbital parameters. The mismatch is computed on the 50000 waveforms on the test set described in the text. On the left plot, we display the mass ratio and the starting frequency $q - f_{\text{min}}$ on the two axes, while we consider the effect of spins on the center and right plot by showing the variables $q - \chi_{1z}$ and $q - \chi_{2z}$, respectively. Each bin is colored according to the *average* mismatch and the three plots share the same color scale. We note that MLGW-SEOBNRV4HM's faithfulness tends to decreases for low values of q, large positive values of s_{1z}, and higher values of f_{min}.

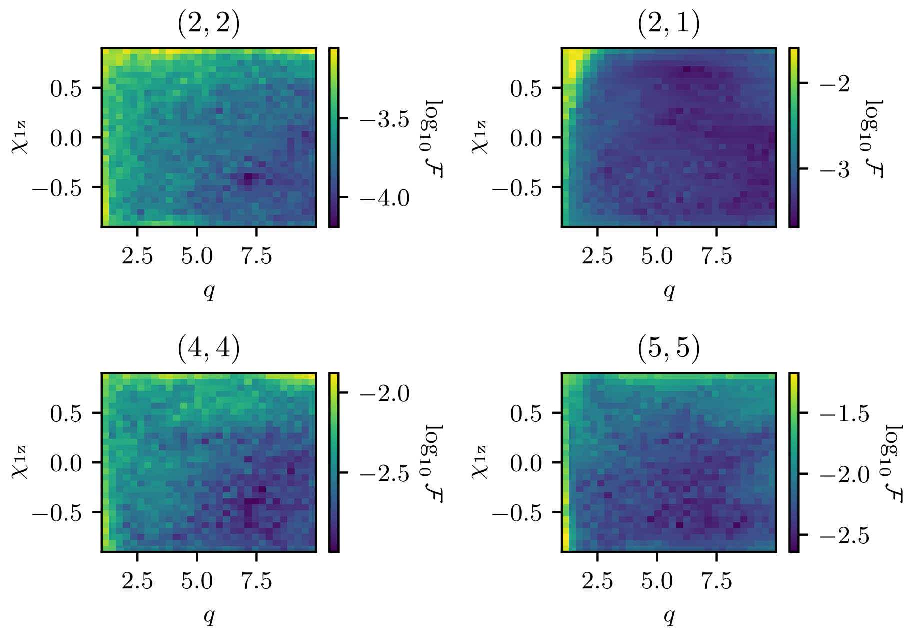

FIG. 4. For each mode, we report the mismatch between MLGW-SEOBNRV4HM and the training model SEOBNRV4HM, as a function of q and s_{1z}. The mismatch is computed on the 50000 waveforms on the test set described in the text. Each bin is colored according to the *average* mismatch. We note that the performance between different modes can vary significantly and in general they decrease for low values of q and high values of spins.

comparison for SEOBNRV4HM_ROM is performed in frequency domain and this involves computing the Fourier tranform of the SEOBNRV4HM waveform. As we perform the comparison in time domain, we omit the latter step, possibly obtaining lower values for the speedup as the ones obtained in [44].

On the other hand, the speedup achieved by MLGW-SEOBNRV4HM is larger than the one obtained by the time

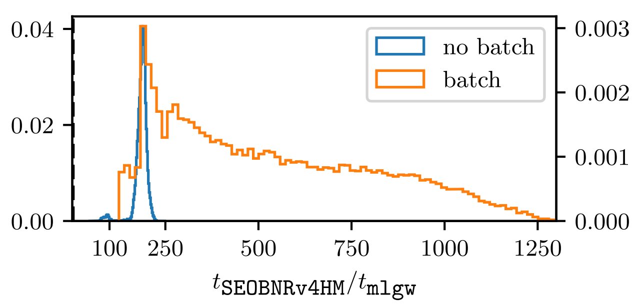

FIG. 5. Speedup provided by MLGW-SEOBNRV4HM over the training model SEOBNRV4HM. In the histogram, we report the ration between the time $t_{\mathrm{SEOBNRv4HM}}$ and the time t_{MLGW} taken by the two models to generate each of the waveforms in the test set. The two histograms are in different scales, reported on the left and right y axis for the "no batch" and "batch" case, respectively. We note that MLGW-SEOBNRV4HM offers a speedup between a 150 and 250 with respect to the training model. MLGW-SEOBNRV4HM offers the option to generate waveforms in batches, effectively parallelizing some linear algebra operations. As shown in the plot, the batch generation provides an additional speedup, which can be as high as 1200.

domain surrogate model SEOBNRV4PHMSUR [45]. Indeed, the authors report a speedup always lower than 100. However, also in this case, the comparison might be biased because the latter study also considers the effects of precession.

IV. FINAL REMARKS AND FUTURE PROSPECTS

Building on our previous work [49,50], we generate a machine learning surrogate model MLGW-SEOBNRV4HM able to reproduce with very high fidelity the output of the widely used approximant SEOBNRV4HM. MLGW-SEOBNRV4HM can generate waveforms in a cuboid $q \times \chi_{1z} \times \chi_{2z} = [1, 10] \times [-0.9, 0.9] \times [-0.9, 0.9]$ on a (reduced) time grid of maximum length of $2s/M_{\odot}$, corresponding to waveforms of length $t \simeq 4.06 \times 10^5 M$ in geometrized units. Our model offers a 2 orders of magnitude speedup over the training model, without trading for accuracy, hence it is an attractive alternative for any data analysis application. To encourage new applications, we release our code (and our trained model) publicly as a Python package through the PyPI repository.[3]

Future work should also include precession. This can be achieved by means of the *spin twist* procedure [32,64–66]. It consist of a time dependent rotation of the plane of emission, resulting in a phase and amplitude modulation which approximates the effect of precession. Training an

[3]The package is distributed under the name MLGW and is available at https://pypi.org/project/mlgw/.

ANN to predict the time dependent rotation is a promising step towards a complete machine learning surrogate model.

While the model is already applicable for most of the parameter estimation problems with current detectors, it is desirable to increase its range of validity, both in parameter space and in time span. In principle, such an extension should be straightforward with the current network setup. On the other hand, due to an increased complexity of the regression task, probably more flexible architectures should be explored, using layers of different size. This would require a more careful (and computationally expensive) hyperparameters tuning.

An enhanced architecture should also benefit from sharing some parameters between models for different HMs—or even from treating the regression of the different modes as a large single regression problem. Indeed, the shapes of the different modes are correlated: for instance, the phases of two HMs are approximately proportional to each other. With the current architecture, the regression for each mode is carried on separately, hence each ANN needs to learn the waveform behavior independently. This could result in many redundant parameters in the network ensemble we introduced here. Inserting parameter sharing inside the regression setup could result in a lighter ANN, which would lead to a reduced inference time.

Finally, we also stress that our PCA + ANN regression framework is fully general and in principle is applicable, with minimal modifications, to any chirplike gravitational wave signal, such as extreme mass ratio inspirals [67,68] or BNSs. Extending the width of parameter space, enriching the BBH model with more physical effects, and supporting a larger variety of systems will become mandatory for the next generation detectors [14,69], when fast and reliable waveform models will be needed to mitigate the huge computational cost posed by very long observed waveforms. Our framework is ideal to achieve such an ambitious goal.

ACKNOWLEDGMENTS

We thank Soumen Roy and Michael Pürrer for useful discussion and their precious comments. S. S. is supported by the research program of the Netherlands Organization for Scientific Research (NWO). This research has made use of data, software, and/or web tools obtained from the Gravitational Wave Open Science Center,[4] 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.

[4] https://www.gw-openscience.org.

[1] B. P. Abbott et al., 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).

[2] R. Abbott et al., 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).

[3] R. Abbott et al., GWTC-2.1: Deep extended catalog of compact binary coalescences observed by LIGO and Virgo during the first half of the third observing run, Phys. Rev. D **109**, 022001 (2024).

[4] R. Abbott et al., GWTC-3: Compact binary coalescences observed by LIGO and Virgo during the second part of the third observing run, Phys. Rev. X **13**, 041039 (2023).

[5] B. P. Abbott et al., Prospects for observing and localizing gravitational-wave transients with Advanced LIGO, Advanced Virgo and KAGRA, Living Rev. Relativity **21**, 3 (2018).

[6] J. Aasi et al., Advanced LIGO, Classical Quantum Gravity **32**, 074001 (2015).

[7] F. Acernese et al., Advanced Virgo: A second-generation interferometric gravitational wave detector, Classical Quantum Gravity **32**, 024001 (2015).

[8] T. Akutsu et al., Overview of KAGRA: Detector design and construction history, Prog. Theor. Exp. Phys. **2021**, 05A101 (2021).

[9] J. Veitch et al., Parameter estimation for compact binaries with ground-based gravitational-wave observations using the LALInference software library, Phys. Rev. D **91**, 042003 (2015).

[10] M. Punturo et al., The Einstein Telescope: A third-generation gravitational wave observatory, Classical Quantum Gravity **27**, 194002 (2010).

[11] M. Maggiore et al., Science case for the Einstein Telescope, J. Cosmol. Astropart. Phys. 03 (2020) 050.

[12] D. Reitze et al., Cosmic Explorer: The U.S. contribution to gravitational-wave astronomy beyond LIGO, Bull. Am. Astron. Soc. **51**, 035 (2019).

[13] M. Evans et al., A horizon study for cosmic explorer: science, observatories, and community, arXiv:2109.09882.

[14] M. Pürrer and C.-J. Haster, Gravitational waveform accuracy requirements for future ground-based detectors, Phys. Rev. Res. **2**, 023151 (2020).

[15] A. Samajdar, J. Janquart, C. Van Den Broeck, and T. Dietrich, Biases in parameter estimation from overlapping gravitational-wave signals in the third-generation detector era, Phys. Rev. D **104**, 044003 (2021).

[16] M. Maggiore, *Gravitational Waves. Vol. 1: Theory and Experiments.*, Oxford Master Series in Physics (Oxford University Press, New York, 2007).

[17] V. Varma, P. Ajith, S. Husa, J. C. Bustillo, M. Hannam, and M. Pürrer, Gravitational-wave observations of binary black holes: Effect of nonquadrupole modes, Phys. Rev. D **90**, 124004 (2014).

[18] V. Varma and P. Ajith, Effects of nonquadrupole modes in the detection and parameter estimation of black hole binaries with nonprecessing spins, Phys. Rev. D **96**, 124024 (2017).

[19] S. Roy, A. S. Sengupta, and K. G. Arun, Unveiling the spectrum of inspiralling binary black holes, Phys. Rev. D **103**, 064012 (2021).

[20] C. Mills and S. Fairhurst, Measuring gravitational-wave higher-order multipoles, Phys. Rev. D **103**, 024042 (2021).

[21] R. Abbott *et al.*, GW190412: Observation of a binary-black-hole coalescence with asymmetric masses, Phys. Rev. D **102**, 043015 (2020).

[22] R. Abbott *et al.*, GW190814: Gravitational waves from the coalescence of a 23 solar mass black hole with a 2.6 solar mass compact object, Astrophys. J. Lett. **896**, L44 (2020).

[23] A. Buonanno and T. Damour, Transition from inspiral to plunge in binary black hole coalescences, Phys. Rev. D **62**, 064015 (2000).

[24] T. Damour and A. Nagar, An improved analytical description of inspiralling and coalescing black-hole binaries, Phys. Rev. D **79**, 081503 (2009).

[25] R. Cotesta, A. Buonanno, A. Bohé, A. Taracchini, I. Hinder, and S. Ossokine, Enriching the symphony of gravitational waves from binary black holes by tuning higher harmonics, Phys. Rev. D **98**, 084028 (2018).

[26] A. Nagar, G. Riemenschneider, G. Pratten, P. Rettegno, and F. Messina, A multipolar effective one body waveform model for spin-aligned black hole binaries, Phys. Rev. D **102**, 024077 (2020).

[27] D. Chiaramello and A. Nagar, Faithful analytical effective-one-body waveform model for spin-aligned, moderately eccentric, coalescing black hole binaries, Phys. Rev. D **101**, 101501 (2020).

[28] S. Ossokine *et al.*, Multipolar effective-one-body waveforms for precessing binary black holes: Construction and validation, Phys. Rev. D **102**, 044055 (2020).

[29] A. Ramos-Buades, A. Buonanno, H. Estellés, M. Khalil, D. P. Mihaylov, S. Ossokine, L. Pompili, and M. Shiferaw, Next generation of accurate and efficient multipolar precessing-spin effective-one-body waveforms for binary black holes, Phys. Rev. D **108**, 124037 (2023).

[30] A. Nagar, A. Bonino, and P. Rettegno, Effective one-body multipolar waveform model for spin-aligned, quasicircular, eccentric, hyperbolic black hole binaries, Phys. Rev. D **103**, 104021 (2021).

[31] S. Khan, S. Husa, M. Hannam, F. Ohme, M. Pürrer, X. Jiménez Forteza, and A. Bohé, Frequency-domain gravitational waves from nonprecessing black-hole binaries. II. A phenomenological model for the advanced detector era, Phys. Rev. D **93**, 044007 (2016).

[32] G. Pratten *et al.*, Computationally efficient models for the dominant and subdominant harmonic modes of precessing binary black holes, Phys. Rev. D **103**, 104056 (2021).

[33] H. Estellés, A. Ramos-Buades, S. Husa, C. García-Quirós, M. Colleoni, L. Haegel, and R. Jaume, IMRPhenomTP: A phenomenological time domain model for dominant quadrupole gravitational wave signal of coalescing binary black holes, Phys. Rev. D **103**, 124060 (2021).

[34] J. Blackman, S. E. Field, C. R. Galley, B. Szilágyi, M. A. Scheel, M. Tiglio, and D. A. Hemberger, Fast and accurate prediction of numerical relativity waveforms from binary black hole coalescences using surrogate models, Phys. Rev. Lett. **115**, 121102 (2015).

[35] V. Varma, S. E. Field, M. A. Scheel, J. Blackman, L. E. Kidder, and H. P. Pfeiffer, Surrogate model of hybridized numerical relativity binary black hole waveforms, Phys. Rev. D **99**, 064045 (2019).

[36] J. Blackman, S. E. Field, M. A. Scheel, C. R. Galley, D. A. Hemberger, P. Schmidt, and R. Smith, A surrogate model of gravitational waveforms from numerical relativity simulations of precessing binary black hole mergers, Phys. Rev. D **95**, 104023 (2017).

[37] J. Blackman, S. E. Field, M. A. Scheel, C. R. Galley, C. D. Ott, M. Boyle, L. E. Kidder, H. P. Pfeiffer, and B. Szilágyi, Numerical relativity waveform surrogate model for generically precessing binary black hole mergers, Phys. Rev. D **96**, 024058 (2017).

[38] V. Varma, S. E. Field, M. A. Scheel, J. Blackman, D. Gerosa, L. C. Stein, L. E. Kidder, and H. P. Pfeiffer, Surrogate models for precessing binary black hole simulations with unequal masses, Phys. Rev. Res. **1**, 033015 (2019).

[39] D. Williams, I. S. Heng, J. Gair, J. A. Clark, and B. Khamesra, Precessing numerical relativity waveform surrogate model for binary black holes: A Gaussian process regression approach, Phys. Rev. D **101**, 063011 (2020).

[40] N. E. Rifat, S. E. Field, G. Khanna, and V. Varma, Surrogate model for gravitational wave signals from comparable and large-mass-ratio black hole binaries, Phys. Rev. D **101**, 081502 (2020).

[41] S. E. Field, C. R. Galley, J. S. Hesthaven, J. Kaye, and M. Tiglio, Fast prediction and evaluation of gravitational waveforms using surrogate models, Phys. Rev. X **4**, 031006 (2014).

[42] M. Pürrer, Frequency domain reduced order models for gravitational waves from aligned-spin compact binaries, Classical Quantum Gravity **31**, 195010 (2014).

[43] M. Pürrer, Frequency domain reduced order model of aligned-spin effective-one-body waveforms with generic mass-ratios and spins, Phys. Rev. D **93**, 064041 (2016).

[44] R. Cotesta, S. Marsat, and M. Pürrer, Frequency domain reduced order model of aligned-spin effective-one-body waveforms with higher-order modes, Phys. Rev. D **101**, 124040 (2020).

[45] B. Gadre, M. Pürrer, S. E. Field, S. Ossokine, and V. Varma, A fully precessing higher-mode surrogate model of effective-one-body waveforms, arXiv:2203.00381.

[46] A. J. K. Chua, C. R. Galley, and M. Vallisneri, Reduced-order modeling with artificial neurons for gravitational-wave inference, Phys. Rev. Lett. **122**, 211101 (2019).

[47] S. Khan and R. Green, Gravitational-wave surrogate models powered by artificial neural networks, Phys. Rev. D **103**, 064015 (2021).

[48] L. M. Thomas, G. Pratten, and P. Schmidt, Accelerating multimodal gravitational waveforms from precessing compact binaries with artificial neural networks, Phys. Rev. D **106**, 104029 (2022).

[49] S. Schmidt, M. Breschi, R. Gamba, G. Pagano, P. Rettegno, G. Riemenschneider, S. Bernuzzi, A. Nagar, and W. Del Pozzo, Machine learning gravitational waves from binary black hole mergers, Phys. Rev. D **103**, 043020 (2021).

[50] J. Tissino, G. Carullo, M. Breschi, R. Gamba, S. Schmidt, and S. Bernuzzi, Combining effective-one-body accuracy and reduced-order-quadrature speed for binary neutron star merger parameter estimation with machine learning, Phys. Rev. D **107**, 084037 (2023).

[51] H. Estellés, M. Colleoni, C. García-Quirós, S. Husa, D. Keitel, M. Mateu-Lucena, M. d. L. Planas, and A. Ramos-Buades, New twists in compact binary waveform modeling: A fast time-domain model for precession, Phys. Rev. D **105**, 084040 (2022).

[52] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, Adaptive mixtures of local experts, Neural Comput. **3**, 79 (1991).

[53] K. Murphy, *Machine Learning: A Probabilistic Perspective*, Adaptive Computation and Machine Learning Series (MIT Press, Cambridge, MA, 2012).

[54] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer-Verlag, Berlin, 2006).

[55] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016), http://www.deeplearningbook.org.

[56] T. Dozat, Incorporating Nesterov Momentum into Adam, in *Proceedings of the 4th International Conference on Learning Representations* (International Conference on Learning Representations, San Juan, Puerto Rico, 2016), pp. 1–4.

[57] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, CoRR abs/1412.6980 (2014).

[58] Hecht-Nielsen, Theory of the backpropagation neural network, in *International 1989 Joint Conference on Neural Networks* (IJCNN: International Joint Conference on Neural Networks, Washington, 1989), Vol. 1, pp. 593–605.

[59] F. Chollet *et al.*, Keras. https://keras.io (2015).

[60] M. Abadi *et al.*, TensorFlow: Large-scale machine learning on heterogeneous systems (2015). Software available from tensorflow.org.

[61] I. Harry, S. Privitera, A. Bohé, and A. Buonanno, Searching for gravitational waves from compact binaries with precessing spins, Phys. Rev. D **94**, 024012 (2016).

[62] I. Harry, J. Calderón Bustillo, and A. Nitz, Searching for the full symphony of black hole binary mergers, Phys. Rev. D **97**, 023004 (2018).

[63] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi *et al.*, Kerastuner. https://github.com/keras-team/keras-tuner (2019).

[64] P. Schmidt, M. Hannam, and S. Husa, Towards models of gravitational waveforms from generic binaries: A simple approximate mapping between precessing and nonprecessing inspiral signals, Phys. Rev. D **86**, 104063 (2012).

[65] P. Schmidt, F. Ohme, and M. Hannam, Towards models of gravitational waveforms from generic binaries II: Modeling precession effects with a single effective precession parameter, Phys. Rev. D **91**, 024043 (2015).

[66] R. Gamba, S. Akçay, S. Bernuzzi, and J. Williams, Effective-one-body waveforms for precessing coalescing compact binaries with post-Newtonian twist, Phys. Rev. D **106**, 024020 (2022).

[67] P. Amaro-Seoane, J. R. Gair, M. Freitag, M. Coleman Miller, I. Mandel, C. J. Cutler, and S. Babak, Astrophysics, detection and science applications of intermediate and extreme mass-ratio inspirals, Classical Quantum Gravity **24**, R113 (2007).

[68] P. Amaro-Seoane, Relativistic dynamics and extreme mass ratio inspirals, Living Rev. Relativity **21**, 4 (2018).

[69] C. B. Owen, C.-J. Haster, S. Perkins, N. J. Cornish, and N. Yunes, Waveform accuracy and systematic uncertainties in current gravitational wave observations, Phys. Rev. D **108**, 044018 (2023).