Gravitational wave modelling with Machine Learning

Stefano Schmidt[1] and Walter Del Pozzo[2]

[1] ????*

[2] *Dipartimento di Fisica "Enrico Fermi", Università di Pisa,
and INFN Sezione di Pisa, Pisa I-56127, Italy*[†]

We apply Machine Learning methods to build a model designed to generate a gravitational waveform in the time domain as produced by a binary black hole coalescence. Our model matches the accuracy of the state-of-the-art EOB models and has a tiny generation cost, equivalents to those of a fast Reduced Order Model. As the generation time does not depend on the lenght of the signal, the model is particularly suitable for long observation surveys, such as ET. Furthermore, it provides a closed form expression for the waveform and its gradient with respect to the orbital parameters. This could lead to a further improvement of the sampling algorithms employed for the parameter estimation.

We train our model on a number of waveforms computed by `TEOBResumS` and we infer a relation between the waveform and the masses m_1, m_2 and (aligned) spins s_1, s_2 of the two BHs. Our implementation is publicly available as a Python package `mlgw` as https://pypi.org/project/mlgw/. `mlgw` has all the features required for performing a full parameter estimation (including the waveform dependence on geometrical parameters). We demonstrate the faithfulness of `mlgw` by successfully reproducing the inference on GW150914 made by the LIGO-Virgo collaboration.

CONTENTS

I. INTRODUCTION

The detection of Gravitational Waves (GW) from compact binary coalescences (CBC) has been possible thanks to the joint effort of a number of different fields of expertise, all joining together to achieve the sophisticated detection process. GW data analysis concerns the detection of a GW signal hidden in the raw detector output (*matched filtering*) and subsequently the inference of its physical properties (*parameter estimation*). In order to accomplish its goal, GW data analysis relies on the availability of waveform (WF) templates to compare with the detector output. To accurately explore the posterior distribution for the parameters defining a CBC, state-of-the-art parameter estimation (PE) algorithms [1] [2] can require the generation of as many as 10^7 waveform templates. It is therefore paramount for the waveform generation to be as fast as possible. At the same time, because of the extreme sensitivity to phase differences in the likelihood function, the templates must retain a high degree of accuracy to avoid biases in the posterior exploration.

Many efforts have been devoted to solve Einstein equations for two coalescing objects and to predict the gravitational radiation emitted []. As solving the full equations is still extremely computationally challenging, the LIGO-Virgo Collaboration relies on approximate analytical models. These can be broadly categorised in two families; (i) the effective-one-body(EOB) waveform models [3] [4]; (ii) the phenomenological models []. EOB models are the most accurate family of analytical models available. They compute the GW signal by solving Hamilton's equations and accurately predict the GW signal up to the plunge. The merger and ringdown parts of the signal are then joint to the inspiral signal using information obtained from numerical relativity. Because of the numerical integrations involved, they tend to be accurate, but slow to generate, see however [] for a much faster approach **(Is this about ROMs?? Or other??)**. The phenomenological waveform are based on the post-Newtonian formalism and then calibrated on EOB waveforms and numerical relativity. They tend to be much faster than EOB models, but not as accurate. Many efforts have been devoted to the task of speeding up the generation of GW signals from EOB families. They have come to be known as *surrogate models*. Surrogate mod-

---

else are constructed starting from some decomposition in the waveform space followed by efficient interpolation to avoid any numerical integration. Being fast to generate, they are routinely employed in GW data analysis.

A Machine Learning model is a promising alternative to the state-of-the-art waveforms generators. Machine Learning (ML) is a branch of statistics that is devoted to reproduce patterns read from data. A ML algorithm needs very little human input and, by a automatically solving an optimization problem, it is able to choose the best performing element among a large class of parametric models for the solution. This is the so-called training procedure. The ML flexibility in modelling data and reproducing trends is appealing: with a proper model choice and with an appropriate training procedure, we can hope to have a reliable, fast to execute generator of GW waveforms, while retaining the accuracy necessary for robust parameter estimation.

In this work, we explore this opportunity. We developed a ML model that is able to generate a GW signal from a binary black hole (BBH) coalescence. We train our model with WFs generated by `TEOBResumS` [5] model and we demonstrate that it can generate GW signals significantly faster than EOB models, matching the performances of a Reduced Order Modelling (ROM) [6] [7] [8]. At the same time it keeps the same accuracy of the underlying training model. Moreover, we present a successful application in a parameter estimation setting, by computing posterior distributions for GW150914.

Our model can be effectively applied to several open problems in GW data analysis. First of all, being among the fastest Inspiral-Merger-Ringdown waveform models and matching the state-of-the-art accuracy, it could be used to speed up the search of GW waves and the subsequent analysis. Furthermore, it is crucial to realise that with our model, the time required to generate a WF does not depend on the lenght of the signal but only on the number of grid points which the WF is evaluated at. Such feature will be crucial for future surveys (e.g. LISA, ET), where "long" signals are expected to be detected, and the problem of the fast generation of a WF will be more pressing.

Finally, our model can be employed for a systematic analysis of the detected signals so far. Being fast to run, it would perform accurate parameter estimation in a low amount of time and it could allow for a comparison between different surrogate WF models[1] , thus yielding invaluable information about the relevant physics.

———————

[1] Indeed it could be trained with different surrogate models: this would create a fast version of the underlying EOB model, which could be employed for parameter estimation.

II. PROBLEM SPECIFICATION

We now provide a precise statement of the objective of our work. A binary black hole is parametrized by a vector $\boldsymbol{\vartheta} = (m_1, m_2, \mathbf{s}_1, \mathbf{s}_2)$, where m_i are the BHs masses and $\mathbf{s}_i = \frac{\mathbf{S}_i}{m_i^2} < 1$ are the *dimensionless* spin. We call them the *orbital parameters*. Let the wave direction of propagation be identified with the spherical coordinates (d_L, ι, φ_0), where d_L is the luminosity distance, ι the polar angle and φ_0 the azimuthal angle. The polar angle ι is also called inclination and it is measured with respect to the normal to the orbital plane. A GW is parametrized as [9]:

$$h(t, d_L, \iota, \varphi_0; \boldsymbol{\vartheta}) = h_+ + ih_\times$$
$$= \frac{1\,\mathrm{Mpc}}{d_L} \sum_{m=-2}^{2} {}^{-2}Y_{2m}(\iota, \varphi_0) \cdot H_{2m}(t; \boldsymbol{\vartheta}) \quad (1)$$

where ${}^{-2}Y_{lm}(\iota, \varphi_0)$ are the spin-2 spherical harmonics. In the expression, and for the rest of this paper, we included only the lowest order term in the multipole expansion, which is the dominant term in every realistic case. We call $(m_1, m_2, \mathbf{s}_1, \mathbf{s}_2, d_L, \iota, \varphi_0)$ *physical parameters* and they fully express the source proprieties as well as its orientation and position. In what follows, we will concentrate on the case in which spins $\mathbf{s}_1$ and $\mathbf{s}_2$ are *aligned*. We plan to extend the experience acquired with this simplified problem to the vastly more complicated case of generic and thus precessing spins.

In the case of aligned spins, Eq. (1) can be further simplified. It turns out that $|H_{2\pm2}| >> |H_{2\pm1}|, |H_{20}|$ and this means that the $|m| \neq 2$ modes of the multipole expansion can be safely neglected. Furthermore, because of parity invariance, it holds $H_{22} = H_{2-2}^*$. Thus a waveform can be fully expressed by the single complex quantity H_{22}. We define:

$$\tilde{h}(t; \boldsymbol{\vartheta}) \equiv {}^{-2}Y_{22}(0,0) \cdot H_{22}(t; \boldsymbol{\vartheta}) = 4 \cdot \sqrt{\frac{5}{64\pi}} H_{22}(t; \boldsymbol{\vartheta}) \quad (2)$$

and we express $\tilde{h}$ in terms of two functions $A(t; \boldsymbol{\vartheta})$ and $\phi(t; \boldsymbol{\vartheta})$ such that:

$$\tilde{h}(t; \boldsymbol{\vartheta}) = A(t; \boldsymbol{\vartheta})e^{i\phi(t; \boldsymbol{\vartheta})} \quad (3)$$

We may also write $f_{\boldsymbol{\vartheta}}(t)$ to denote a function $f(t; \boldsymbol{\vartheta})$ of time with parametric dependence on $\boldsymbol{\vartheta}$.

With this definition, the full waveform can be expressed as:

$$h(t, d_L, \iota, \varphi_0; \boldsymbol{\vartheta}) =$$
$$= \frac{1\,\mathrm{Mpc}}{d_L} \cdot \left\{ \frac{1 + \cos^2 \iota}{2} \cdot A_{\boldsymbol{\vartheta}}(t) \cos[\phi_{\boldsymbol{\vartheta}}(t) + 2\varphi_0] \right.$$
$$\left. + i \cdot \cos \iota \cdot A_{\boldsymbol{\vartheta}}(t) \sin[\phi_{\boldsymbol{\vartheta}}(t) + 2\varphi_0] \right\} \quad (4)$$

where we split the real and the imaginary part of h and we used the relation $^{-2}Y_{2\pm2}\left(\iota,\varphi_0\right)=(1\pm\cos\iota)^2e^{\pm i2\varphi_0}$. We choose the convention that $\phi_{\boldsymbol{\vartheta}}=0$ when the amplitude $A_{\boldsymbol{\vartheta}}$ has a maximum [2].

The goal of the present work is to provide an accurate ML model which outputs the functions A and ϕ, as generated by the state-of-the-art time domain WF models.

Apart from a numerical constant, the quantity $\tilde{h}(t;\boldsymbol{\vartheta})=A(t;\boldsymbol{\vartheta})e^{i\phi(t;\boldsymbol{\vartheta})}$ is the $h=h_++ih_\times$ strain, generated by a BBH with orbital parameters $\boldsymbol{\vartheta}$, when observed from the direction orthogonal to the total angular momentum ($\iota=0$), at a luminosity distance $d_L=1\,\mathrm{Mpc}$ and with $\varphi_0=0$. $A(t)$ represent the amplitude of $\tilde{h}_+$ as a function of time, while $\phi(t)$ represent the phase of $\tilde{h}_+$ as a function of time.[3]

The set of independent orbital parameters to consider is $\boldsymbol{\vartheta}=(m_1,m_2,s_1,s_2)$ where s_i is the spin magnitude of the i-th BH. It is convenient to parametrize the two masses with the mass ratio q and the total mass M:

$$q=\frac{m_1}{m_2}>1 \qquad M=m_1+m_2 \tag{5}$$

with the convention that $m_1>m_2$.

We now state our problem precisely. We wish to find a ML model that reliably represents the following map:

$$(q,M,s_1,s_2)\longmapsto A_{(q,M,s_1,s_2)}(t) \tag{6}$$
$$(q,M,s_1,s_2)\longmapsto \phi_{(q,M,s_1,s_2)}(t) \tag{7}$$

where $A(t)$ and $\phi(t)$ are as in equation (3).

III. THE MODEL

A. Modelling dependence on total mass

Before developing the full ML model, the problem (6, 7) can be further simplified using the global scale invariance of general relativity in vacuum. This will allow us to model analytically the dependence on total mass M.

Following [10], we recognize that the free theory lacks of a built in scale. For this reason, a scale coordinate transformation $x^\mu\to\lambda x^\mu$ must not affect the physics, as long as the other quantities are scaled accordingly. The scalings are as follows:

———————

[2] Indeed, a constant translation of $\phi_{\boldsymbol{\vartheta}}$ can be absorbed in the definition of φ_0 and does not affect the physics.
[3] We could have also worked with $\mathrm{Re}\tilde{h}$ and $\mathrm{Im}\tilde{h}$. However, such choice is most prone to numerical inaccuracies: the strain is a fast oscillating function and tracking its behavior is difficult.
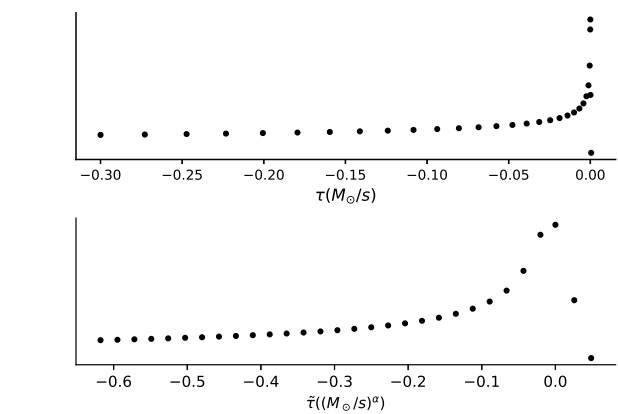

FIG. 1. Amplitude (arbitrary units) of a GW wave represented on a grid with 30 points. On left panel, amplitude values are plotted in the reduced time grid $\boldsymbol{\tau}$ (units in $\frac{s}{M_\odot}$). On the right panel, the same function is represented as a function of the transformed grid $\tilde{\boldsymbol{\tau}}$ (units in $\left(\frac{s}{M_\odot}\right)^\alpha$). It is set $\alpha=0.4$. It is manifest that $\boldsymbol{\tau}$ grid is finer around the merger. On the other hand in the transformed grid $\tilde{\boldsymbol{\tau}}$, amplitude has a smoother behaviour.

$x\to\lambda x$	(distance)	$t\to\lambda t$	(time)
$M_i\to\lambda M_i$	(mass)	$p_i\to\lambda p_i$	(momentum)
$J_i\to\lambda J_i$	(angular momentum)	$s_i\to s_i$	(dimensionless spin)
$E\to\lambda E$	(energy)	$f\to\lambda^{-1}f$	(frequency)
$h\to h$	(strain)	$v\to v$	(velocity)

This can be applied to the case of GW generated by a BBH coalescence to obtain the analytical total mass dependence of amplitude $A_{\boldsymbol{\vartheta}}$ and phase $\phi_{\boldsymbol{\vartheta}}$:

$$\frac{1}{M'}A_{(q,M',s_1,s_2)}\left(\frac{t}{M'}\right)=\frac{1}{M}A_{(q,M,s_1,s_2)}\left(\frac{t}{M}\right) \tag{8}$$

$$\phi_{(q,M',s_1,s_2)}\left(\frac{t}{M'}\right)=\phi_{(q,M,s_1,s_2)}\left(\frac{t}{M}\right) \tag{9}$$

where M, M' are arbitrary total mass.

Thus, in a ML model we only need to consider the parameters $\tilde{\boldsymbol{\vartheta}}=(q,s_1,s_2)$. Once such model is fitted, one can use equations (8) and (9) to produce a waveform with arbitrary total mass.

B. The time grid

The first task to build a waveform generator is to choose a representation of the waveforms. Each function to fit f (i.e. amplitude and phase) must be represented by its values $\mathbf{f}\in\mathbb{R}^D$ on a discrete grid of D points $\mathbf{t}\in\mathbb{R}^D$. In equations (8) and (9), only dimensionless time $\tau=\frac{t}{M}$ appears and it is then convenient to work in a grid of reduced time $\boldsymbol{\tau}$. The time grid is chosen

with the convention that at $\tau = 0$ the function $A(t; \boldsymbol{\vartheta})$ (i.e. the amplitude of the H_{22} mode) has a peak.

Both amplitude and phase are represented by:

$$\mathbf{f}(\tilde{\boldsymbol{\vartheta}})_i = f_{\tilde{\boldsymbol{\vartheta}}}(\boldsymbol{\tau}_i) \tag{10}$$

where f stands for any of the functions $A_{\tilde{\boldsymbol{\vartheta}}}(t)$ and $\phi_{\tilde{\boldsymbol{\vartheta}}}(t)$.

The value of f at an arbitrary time must be found by interpolation and introduces an error in the value of the function. To make the interpolation effective, we want a finer grid when the function changes much. Clearly an equally spaced grid over times is not the best choice since the amplitude has a very narrow peak at $\tau = 0$ [4]. A good solution is to create an equally spaced grid $\tilde{\boldsymbol{\tau}}$ for the variable

$$\tilde{\tau} \equiv \operatorname{sign} \tau \cdot (|\tau|)^{\alpha}; \quad \alpha < 1 \tag{11}$$

and build the τ grid $\boldsymbol{\tau}$ as:

$$\boldsymbol{\tau}_i = \operatorname{sign} \tilde{\boldsymbol{\tau}}_i \cdot (|\tilde{\boldsymbol{\tau}}_i|)^{\frac{1}{\alpha}} \tag{12}$$

We call α *distortion parameter*. As can be seen in figure 1, this choice ensures that more points are accumulated around the amplitude peak.

In this framework, our problem reduces to performing the regression:

$$\tilde{\boldsymbol{\vartheta}} = (q, s_1, s_2) \in \mathcal{P} \longmapsto \boldsymbol{A}_{\tilde{\boldsymbol{\vartheta}}} \in \mathbb{R}^D \tag{13}$$

$$\tilde{\boldsymbol{\vartheta}} = (q, s_1, s_2) \in \mathcal{P} \longmapsto \boldsymbol{\phi}_{\tilde{\boldsymbol{\vartheta}}} \in \mathbb{R}^D \tag{14}$$

where $\mathcal{P} \subset \mathbb{R}^3$ is the domain in which the physical parameters are allowed to vary. Spins must be in range $[-1, 1]$ and the mass ratio must satisfy $q > 1$. Usually, the training EOB models are calibrated for domains $\mathcal{P} \simeq [1, 20] \times [-0.9, 0.9] \times [-0.9, 0.9]$. A conservative choice leads to use a subset of the full calibration domain.

C. Dataset creation

As in any ML method, we must create a dataset in order to perform training. For our model, the dataset $X \in \mathbf{Mat}(N, 3 + 2D)$ of N waveform has the following form:

$$X_{i:} = [q, s_1, s_2, A_{\tilde{\boldsymbol{\vartheta}}}^T, \boldsymbol{\phi}_{\tilde{\boldsymbol{\vartheta}}}^T] \tag{15}$$

where $X_{i:}$ denotes the i-th row of the dataset matrix.

The dataset is filled with random values of parameters $\tilde{\boldsymbol{\vartheta}}_i \sim \operatorname{Unif}(\mathcal{P})$ [5]. To generate the training waves the

TEOBResumS model is used **(CITE...)**. Waves are generated with a standard set of parameters: $\iota = 0$, $\varphi_0 = 0$, $d_0 = 1 \, \mathrm{Mpc}$ and $M = 20 \, \mathrm{M}_{\odot}$. The output of the training surrogate model must be interpolated to the chosen grid.

It is important to ensure that all waves have zero phase at a constant time point $\bar{t}$. This is crucial in order to have a continuous dependence of the phase components on the orbital parameters; a random alignment would wash out any relation. Model performances are not seen to depend on the choice of $\bar{t}$.

D. Dimensionality reduction

Once we are able to represent waveforms, a regressions $\tilde{\boldsymbol{\vartheta}} \longmapsto A_{\tilde{\boldsymbol{\vartheta}}}, \phi_{\tilde{\boldsymbol{\vartheta}}}$ is unfeasible. The dimension of the target space is too large for this task. Luckily, the elements of $\boldsymbol{A}, \boldsymbol{\phi}$ are strongly correlated with each other: the independent amount of information, required to fully reconstruct the wave, can be stored in a low dimensional vector. A number of ML techniques to perform such a task are available. Among them, Principal Component Analysis (PCA) [11, ch. 12] was found to be particularly effective.

A PCA model is trained with the dataset in (15): it represents an (approximate) bijective map between the high dimensional waveform $\mathbf{f} = \boldsymbol{A}, \boldsymbol{\phi} \in \mathbb{R}^D$ and the low-dimensional representation $\mathbf{g} = \mathbf{g}_A, \mathbf{g}_{\phi} \in \mathbb{R}^K$. Here K is the dimension of low-dimensional representation. In what follows, $\mathbf{f}$ will be the high order representation of any of the function $A_{\tilde{\boldsymbol{\vartheta}}}(t)$, $\phi_{\tilde{\boldsymbol{\vartheta}}}(t)$, while $\mathbf{g}$ will be the PCA reduced lower order representation of $\mathbf{f}$. The relation takes the following form:

$$\mathbf{g} = H(\mathbf{f} - \boldsymbol{\mu}) \tag{16}$$

$$\mathbf{f} = H^T \mathbf{g} + \boldsymbol{\mu} \tag{17}$$

where, as usual in PCA, $H \in \mathbf{Mat}(K, D)$ is the matrix whose colums are the first K eigenvalues of the empirical covariance matrix $\Sigma = \frac{1}{N} \sum_{i=1}^{N} \mathbf{f}_i \mathbf{f}_i^T \in \mathbf{Mat}(D, D)$; the eigenvectors are also called principal components of the data (PCs). $\boldsymbol{\mu}$ is the empirical mean vector: $\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{f}_i)$. In the equations above, the $\tilde{\boldsymbol{\vartheta}}$ dependence is omitted for notational simplicity.

E. Regression

Once a dimensional reduction (and reconstruction) scheme is available, the aim is to perform the regression

$$\tilde{\boldsymbol{\vartheta}} \longmapsto \boldsymbol{g}(\tilde{\boldsymbol{\vartheta}}) \tag{18}$$

A number of ML models are avaiable for this purpose. The model Mixture of Experts (MoE) [12] [11, ch. 11] is found to be a good choice.

---

[4] As the phase has a rather regular behavior, it is not important to choose the right grid. For this reason a single grid for amplitude and phase, tuned on the behavior of amplitude, is used.

[5] We denote by $\operatorname{Unif}(\mathcal{P})$ a uniform probability distribution on the set $\mathcal{P}$.

It performs the following 1D regression:

$$y(\mathbf{x}) = \sum_{l=1}^{L}(W^T\mathbf{x})_l \cdot \mathcal{S}(V^T\mathbf{x})_l \qquad (19)$$

where $\mathcal{S}$ is the *softmax function*:

$$\mathcal{S}(V^T\mathbf{x})_l = \frac{e^{(V^T\mathbf{x})_l}}{\sum_{l'=1}^{L} e^{(V^T\mathbf{x})_{l'}}} \qquad (20)$$

and $\mathbf{x} \in \mathbb{R}^{\tilde{M}}$ and $V, W \in \mathbf{Mat}(\tilde{M}, L)$. The meaning of eq. (19) is clear: the output is a weighted combination of L linear regressions $(W^T\mathbf{x})_l$ (each called an *experts*); each expert perform a reliable regression in a small region of the space. The softmax function (in this context also called *gating function*) switches on the expert contributions whenever this is required. MoE is usually fitted with the Expectation Maximization (EM) algorithm, which iteratively sets the W and V by refining a lower bound to the log-likelihood of the model.

Linear regression is a very simple model, often inadequate to model a complex relation. A simple trick to improve its performance is called *basis functions expansion*. It consist in the replacement:

$$\mathbf{x} \longrightarrow \boldsymbol{\xi}(\mathbf{x}) = [\xi_1(\mathbf{x}), \ldots, \xi_M(\mathbf{x})]^T \qquad (21)$$

Thus, each expert becomes a non linear regression of the input $\mathbf{x}$. A careful choice of basis functions can really make a difference in fit performances and it must be done at validation time, by comparing performances of different models.

In our model, including in the ξ_i every monomial up to $3/4$th order in the three variables $(\log q, s_1, s_2)$ is a good working choice [6].

The MoE is fitted with a reduced dimensional version of dataset (15):

$$G_{i:} = [q, s_1, s_2, \mathbf{g}_A(\tilde{\boldsymbol{\vartheta}})^T, \mathbf{g}_\phi(\tilde{\boldsymbol{\vartheta}})^T] \qquad (22)$$

The user must choose the number L of experts to use and the basis functions features $\boldsymbol{\xi}(\tilde{\boldsymbol{\vartheta}}) \in \mathbb{R}^M$ to use.

As MoE model deals with single dimensional outputs, a single independent regression must be performed for each component g_k of $\mathbf{g} \in \mathbb{R}^K$ [7]. In general, a regression will be a collection of MoE weights $\{W^{(k)}, V^{(k)} \in \mathbf{Mat}(M, L_k)\}_{k=0}^{K}$, where index k labels different regressions for each PC.

---

[6] The choice of working with variable $\log q$ rather than q gives much better validation results. Heuristically, using $\log q$ prevents the values of the data features to vary too much within the range of interest, yielding more stable numerical performance.

[7] This is not a great limitation, because, due to orthogonality of PCs, each g_j is independent from the other: we do not miss correlation among different regressions.

F. Summary

The model has the following explicit form:

$$\mathrm{model} : \mathcal{P} \subset \mathbb{R}^3 \to \mathbb{R}^K \to \mathbb{R}^D$$

$$\tilde{\boldsymbol{\vartheta}} \longmapsto \mathbf{g}(\tilde{\boldsymbol{\vartheta}}) = \begin{pmatrix} \sum_{l=1}^{L_1}(W^{(1)\,T}\boldsymbol{\xi})_l \cdot \mathcal{S}(V^{(1)\,T}\boldsymbol{\xi})_l \\ \vdots \\ \sum_{l=1}^{L_K}(W^{(K)\,T}\boldsymbol{\xi})_l \cdot \mathcal{S}(V^{(K)\,T}\boldsymbol{\xi})_l \end{pmatrix}$$

$$\longmapsto \mathbf{f}(\tilde{\boldsymbol{\vartheta}}) = H^T\mathbf{g}(\tilde{\boldsymbol{\vartheta}}) + \boldsymbol{\mu} \qquad (23)$$

where $\boldsymbol{\xi}(\tilde{\boldsymbol{\vartheta}}) \in \mathbb{R}^M$ are the chosen basis function for the regression and $\mathcal{S}(\cdot)_k$ is the *softmax* function eq (20). Two relations of the same type must be fitted, one for the amplitude, the other for the phase.

Once weights are set properly, the expression provides an estimation for the waveform $\tilde{h}$ in (2) and (3). The fitted expression for $\tilde{h}$ is evaluated at constant mass $M = 20\,\mathrm{M}_\odot$; the dependence on total mass must be included with equations (8) and (9). The dependence on (d_L, ι, φ_0) is computed with eq. (4). With this method, we are able to obtain a complex waveform $h(t; m_1, m_2, s_1, s_2, d_L, \iota, \varphi_0)$ which reproduces closely a waveform from the training surrogate model.

A Python implementation of the model is released in the package `mlgw`, publicly available at pypi.org/project/mlgw/. The user can install it with the command `pip install mlgw`. The package provides also the gradients $\frac{\partial h}{\partial \vartheta_i}$ of the waveform.

The model can generate a waveform also for $\tilde{\boldsymbol{\vartheta}} \notin \mathcal{P}$, without guarantee to provide a reliable result. The waveform can be evaluated on a user given time grid by means of interpolation; the grid must ensure that the peak of amplitude happens at $t = 0$. If the user asks for an extrapolation outside the dataset time grid, the model returns $h = 0$.

Roughly, the minimum frequency in the signal as a function of M, q and τ_{min} is given by:

$$f_{min} = 151\,\mathrm{Hz} \left(\frac{(1+q)^2}{q}\right)^{\frac{3}{8}} \left(\frac{M_\odot}{M}\right) \left(\frac{1\,\frac{\mathrm{s}}{\mathrm{M}_\odot}}{\tau_{min}}\right)^{\frac{3}{8}} \qquad (24)$$

The expression is approximate because it is obtained within a Newtonian framework and does not consider spin effects.

IV. MODEL PERFORMANCE

We now discuss some tests on our model. We first study how its performance depends on the choice of hyperparameters. Furthermore, we assess the model accuracy and its limitations. Finally, we measure the speed up provided by our model as compared with standard EOB methods. As it is common, we measure the similarity
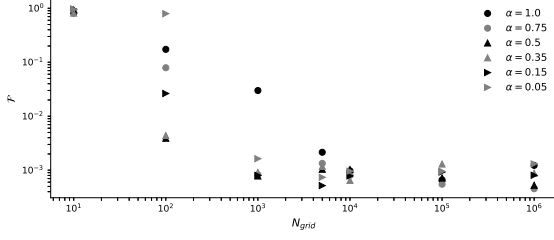
FIG. 2. Mismatch between waves $\mathbf{f}_{N_{grid},\alpha}$ and raw waves from EOB model, as a function of time grid size N_{grid}. Each series refers to a different values of α. Mismatch is computed on 10 test waves.

between two waves by means of the *optimal mismatch*:

$$\mathcal{F}[h_1, h_2] \equiv \underset{\phi_0}{\arg\min} \left[1 - \frac{\langle h_1, h_2 e^{i\phi_0} \rangle}{\sqrt{\langle h_1, h_1 \rangle \langle h_2, h_2 \rangle}} \right] \qquad (25)$$

$$= \underset{\phi_0}{\arg\min} \left[1 - \frac{\mathrm{Re} \int \mathrm{d}f \, \frac{\tilde{A}_1 \tilde{A}_2}{S_n} e^{i(\phi_1 - \phi_2 - \phi_0)}}{\sqrt{\left(\mathrm{Re} \int \mathrm{d}f \, \frac{\tilde{A}_1^2}{S_n} \right) \left(\mathrm{Re} \int \mathrm{d}f \, \frac{\tilde{A}_2^2}{S_n} \right)}} \right]$$

A. Validation

Wherever relevant, we will employ a dataset with 5800 waveforms generated in the domain $\mathcal{P} = [1, 20] \times [-0.8, 0.95] \times [-0.8, 0.95]$, with $\tau_{min} = 1.0\,\mathrm{s/M_\odot}$. The dataset was generated with TEOBResumS model [].

a. *Dataset generation parameters* We first evaluate the impact of number of grid points N_{grid} and distortion parameter α. Let $\mathbf{f}_{N_{grid},\alpha}$ the wave stored in a dataset where τ_{min} and $\mathcal{P}$ are fixed as above. We compare it with the output of the EOB model $\mathbf{f}_{EOB}$. To compare the two waves, $\mathbf{f}_{N_{grid},\alpha}$ must be evaluated on the dense time grid of EOB, by means of linear interpolation. We then vary N_{grid} and α and compute the resulting mismatch $\mathcal{F}[\mathbf{f}_{EOB}, \mathbf{f}_{N_{grid},\alpha}]$. We report the results in figure 2.

As expected, we note that, by increasing the number of grid points, the mismatch decreases. Furthermore, using more than $\sim 10^3$ grid points, does not bring any improvement to mismatch. The result is dominated by numerical errors in the interpolations and it provides a lower-bound for the performances of the fit. We note that a dataset grid with $\alpha \simeq 0.35 - 0.5$ is a good choice. It is effective if N_{grid} does not grow too much. For a high number of grid points, different values of α yield almost equivalent results. A good setting for dataset hyperparameters could be: $N_{grid} \simeq 2/3 \cdot 10^3$ and $\alpha \simeq 0.3/0.5$.

b. *MoE parameters* In what follows, we will focus on setting a small number of hyperparameters relevant to MoE model: the number of experts N_{exp} for each component model, the basis functions $\xi_i(\tilde{\boldsymbol{\vartheta}})$ to use in basis function expansion and the number N_{train} of training points to use. Other parameters, related to the details of the training procedure, will not be considered.
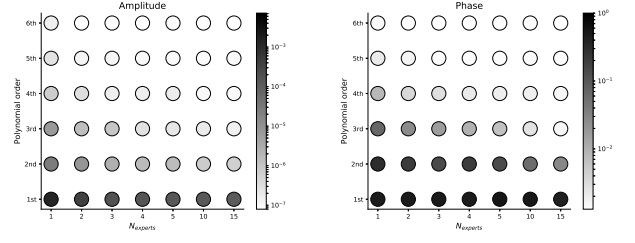

FIG. 3. Validation results for fit of MoE model. Each point corresponds to a MoE regressions for the amplitude (left) and phase (right), with a different values of expert number N_{exp} and order of polynomial basis function. The amplitude and phase are represented with 5 and 4 PCs respectively. In the colorbar, we represent the mismatch on test waves: it is obtained by reconstructing test waves with fitted amplitude (phase) and test phase (amplitude).
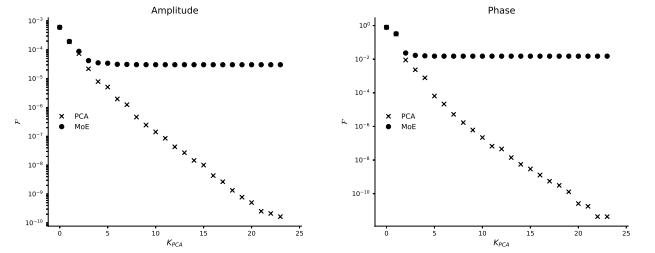

FIG. 4. Test mismatch as a function of the number of PCs used in the low dimensional representation. Label "PCA" refers to waves reconstructed with PCA only; points with label "MoE" are reconstructed after a MoE regression. Data refers to amplitude (left panel) and phase (right panel). MoE model is chosen to be the optimal one with 4 experts and a fourth order polynomial.

In figure 3 we show our results. We fitted a model for amplitude (or phase) for different configurations of expert number N_{exp} and polynomial basis function. By label "n-th order", we mean that in the basis function expansion, every monomial up to n-th order is used. We report with a colorbar the value of the mismatch F between test and reconstructed WFs. The MoE models for each component share the same number of experts N_{exp}. The test mismatch for the fitted amplitude (phase) is computed by using the test phase (amplitude) in the reconstructed wave.

Interpretation of the results is straightforward. Fit performances improves, as the model complexity (i.e. number of fittable parameters) increases. In general, we note that adding more features (i.e. improving expert's flexibility) is more effective than increasing the number of experts. However, model performances do not improve indefinitely. As model complexity increases beyond a threshold, performance does not get any better. Such threshold is around a model with 4 experts and 4th order polynomial regression. This "simple" model match the performances of much more complex models and thus it should be deemed as the best choice.

c.   Choosing the number of PCs   Of course, the accuracy of the reconstruction of the low dimensional representation depends on the number K of principal components considered: the more PCs are used, the best accuracy can be achieved. However in practice, due to error in the MoE regression, one cannot reduce the reconstruction mismatch arbitrarily and should not choose the number of PCs to fit before having seen the MoE accuracy. At high PC order the relations to fit become noisy and the regression becomes less accurate, eventually washing out any improvement brought by a higher number of PCs.

In figure 4 we report a numerical study of this. We plot the reconstruction mismatch as a function of the number of PCs considered. We consider separetely regression for the amplitude and for the phase. In one series, we reconstruct the wave using true values of PCs: the mismatch is a measure of PCA accuracy. In the other series, we reconstruct a wave using values for PCs as guessed by MoE regression: this is a measure of accuracy of both PCA and regression. For the first two PCs, the regression is accurate enough for reproducing the PCA accuracy. On the other hand, any regression beyond the 4th PCA component does not give any improvement to the MoE mismatch: the noise in the relation of high order PCs is too high for a regression to be performed.

In the PCA, we include every PC which yields improvement in MoE mismatch. For our model, $K = 5(4)$ is a good choice for amplitude (phase). Of course, this strongly depends on the regression model: the more precise the model is, the more PCs can be included. However, any model cannot increase its accuracy indefinitely. Every surrogate model has an intrinsic noise level, due to numerical error and to the approximations in the physical model.

d.   Choosing the number of training points   The choice of the number of training points N_{train} must trade between accuracy and speed of execution. Too many training points will make the training slow and unfeasible, while too few training points will yield a poor model, which does not generalize data (underfitting). In the choice of number of training points, the comparison between train and test error will provide important information on how the model is able to generalize the trend.

In figure 5 we report train and test value of mismatch and mse of first 3 PCs as a function of the number of training points. Data refers to a MoE model fitted for 4 PCs of the phase dataset. The models has $N_{exp} = 4$ and performs basis function expansion with a fourth degree polynomial. Test mismatch are obtained using test amplitude to reconstruct the waveform; this is not a great limitation as any error in phase reconstruction dominates the overall mismatch (see e.g. fig. 4).

As N_{train} increases, we see a steady decrease of the errors, until a plateau is reached. We note that overfitting is not a problem. For a reasonably high number of training points ($N_{train} \gtrsim 50$) train and test error are close to each other. For $N_{train} \gtrsim 800$, the trend stabilises and in-
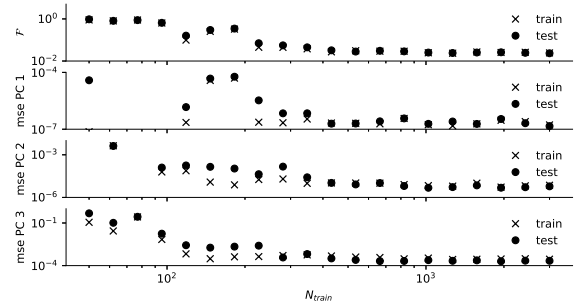

FIG. 5. Train and test error for MoE fit of 4 PCs of phase, as a function of the number of training points. We report train and test reconstruction mismatch (top) and mse for the first 3 PCs (below). MoE model employs 4 experts and a fourth order polynomial for a basis function expansion.
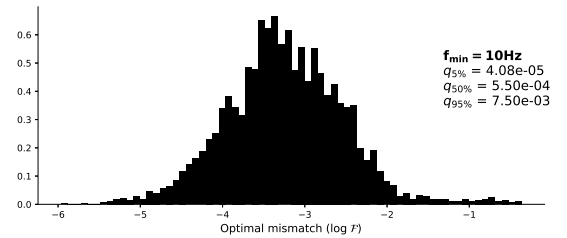

FIG. 6. Histogram for the logarithm of mismatch values, computed on $N = 4000$ test waveforms. Each WF is generated with random orbital parameters ($m_1, m_2, s_1, s_2, d_L, \iota, \phi_0$) and with a starting frequency of $10\,\text{Hz}$. We report the median value $q_{50\%}$ as well as the positions $q_{5\%}$ and $q_{95\%}$ of the 5th and 95th percentile. The 95% of the test waves have a mismatch lower than $7.5 \cdot 10^{-3}$.

creasing training points does not affect much model performance. In the present model, setting $N_{train} \simeq 3000$ is a good choice [8].

B.   Accuracy

a.   Test mismatch   We compute mismatch value on a number of randomly generated waves. The mismatch is computed with a flat noise curve, i.e. a constant power spectral density for the detector noise. In order to tests the model in the most general case, we extract a random values of the physical parameters (i.e. $m_1, m_2, s_1, s_2, d_L, \iota, \varphi_0$) for each wave. We report our results in the histogram in figure 6. We report a median value of the distribution $\mathcal{F}_m = 4 \cdot 10^{-3}{}^{+???}_{-???}$. Such results

___

[8] As compared with standard neural networks, which routinely employ $O(10^5)$ points datasets, this is an incredibly low amount of data. This is due to the fact that MoE is a simple model with a few number of parameters: few data are enough for learning a reliable relation.

are similar to the discrepancies between state-of-the-art models.

To understand better model performances, it is interesting to display the accuracy as a function of the orbital parameters $\vartheta = (q, M, s_1, s_2)$. We generate waves for randomly chosen values of $\vartheta = (q, M, s_1, s_2)$ and, for each wave, we measure test mismatch $\mathcal{F}$ and mse on the reconstruction of the first PC for the phase. The latter is useful to test the accuracy of the fit before wave reconstruction. The results are reported in the contour plots in figure 9.

By looking at the top line of 9, we note that the mse does not depend on M. This was to be expected because the total mass dependence is inserted analytically within the model.

Again on the top line, we note that both $\mathcal{F}$ and mse depend (quite strongly) on q. The mismatch and mse are well correlated: the reconstruction of the wave and the interpolation do not affect much the mismatch. We note that for low values of q the fit shows poor performance: in that region the relation to catch is less smooth, making the regression less reliable.

In the center line of 9, we displayed the dependence on spins. As long as the s_1 dependence is considered, the most striking feature is the inverse correlation of mismatch and mse for the first phase PC. This means that the value of the first PC does not affect much the spin dependence of the waveform and that, being non-leading, spin contribution become important at lower order of PCs. Indeed, the values of the first PC are well correlated with mismatch in the case of q. See [13] for a closely related discussion on PCA components and its dependence on physical parameter. **Is it fine to cite this? Is it relevant?**

Apart from this feature, we note that the mismatch tends to grow for high spins: the fit is less reliable in such regions.

In the third row of 9, we displayed the dependence on q and s_1, the variables on which the error depends more. We note again the inverse correlation between the mismatch and mse, when considering the s_1 dependence.

b. *Parameter estimation* We run a PE on the signal GW150914 [14]. We performed a sampling from the posterior of the signal with a Nested Sampling algorithm [15] [1]. Besides a set of sampled points, the algorithm provides an estimation of the Bayes factor for the model[9]. We plot some (marginalized) posterior probability distribution functions (PDF) in figures 7. In table I, we report our physical predictions for the source parameters and a number of other quantities and we compare them with the published LIGO-Virgo results in [16].

---

[9] The Bayes factor $\mathcal{B} = \frac{Z_z}{Z_n}$ is the ratio between the evidence of the model Z_s, where data are assumed to be a superposition of GW signal and noise, and the evidence Z_n of the noise model, where data are assumed to be composed only by noise. It measures framework the probability that the WF model explain the data.

	LIGO-Virgo	mlgw
Total mass $\mathcal{M}_c(M_\odot)$	$70.3^{+5.3}_{-4.8}$	$74.2^{+4.5}_{-4.8}$
Chirp mass $\mathcal{M}_c(M_\odot)$	$30.2^{+2.5}_{-1.9}$	$32.0^{+1.9}_{-2.4}$
Primary mass $m_1(M_\odot)$	$39.4^{+5.5}_{-4.9}$	$40.6^{+5.4}_{-4.0}$
Secondary mass $m_2(M_\odot)$	$30.9^{+4.8}_{-4.4}$	$33.6^{+5.4}_{-4.1}$
Inverse mass ratio $\bar{q} = \frac{1}{q} = \frac{m_2}{m_1}$	$0.79^{+0.18}_{-0.19}$	$0.83^{+0.15}_{-0.21}$
Effective spin parameter χ_{eff}	$-0.09^{+0.19}_{-0.17}$	$-0.08^{+0.13}_{-0.17}$
(dimensionless) spin s_1	$0.32^{+0.45}_{-0.28}$	$0.14^{+0.51}_{-0.66}$
(dimensionless) spin s_2	$0.57^{+0.40}_{-0.51}$	$-0.02^{+0.78}_{-0.59}$
Luminosity distance $d_L(Mpc)$	390^{+170}_{-180}	490^{+158}_{-208}
Final mass $M_f(M_\odot)$	$67.1^{+4.6}_{-4.4}$	$70.5^{+3.9}_{-4.2}$
Final spin $s_L(M_\odot)$	$0.67^{+0.06}_{-0.08}$	$0.70^{+0.05}_{-0.08}$
Log Bayes factor $\log\mathcal{B} = \log\frac{Z_s}{Z_n}$	$288.7^{+0.2}_{-0.2}$	$302^{+0.2}_{-0.2}$

TABLE I. Several physical quantities extracted from signal GW150914 by sampling the posterior distribution. Results of `mlgw` model are compared by those published by the LIGO-Virgo collaboration (using EOB model) [17]. We report the measures (i.e. the median value of the marginalized posterior) and their 90% confidence interval. The final BH mass and spins are obtained with the formula from [18].
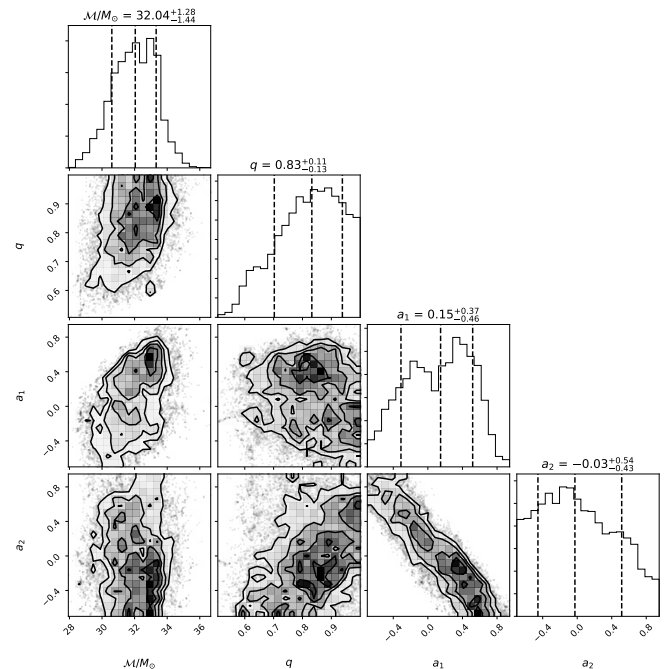

FIG. 7. Marginalized posteriors for the chirp mass $\mathcal{M}_c$, inverse mass ration $\bar{q} = \frac{1}{q}$ and spins s_1, s_2 for GW150914. We report the histograms for the one dimensional posteriors of each quantity; the dashed lines reports the median value as well as the 68% credible interval. In the contour plots, we report the two dimensional posteriors for each pair of quantities. The figure holds 16411 samples and is generated with Python package `corner` [19].
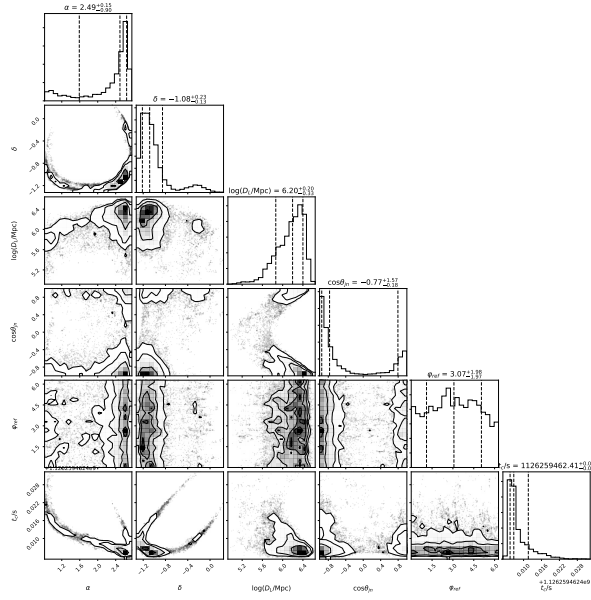
FIG. 8. Marginalized posteriors for the sky position angles, luminosity distance, time of coalescence and cosine of the inclination of the orbital plane for GW150914. We report the histograms for the one dimensional posteriors of each quantity; the dashed lines reports the median value as well as the 68% credible interval. In the contour plots, we report the two dimensional posteriors for each pair of quantities. The figure holds 16411 samples and is generated with Python package `corner` [19].

We note that the predictions of `mlgw` are consistent with those of the LIGO-Virgo collaboration. The mass related parameters (i.e. chirp mass, total mass and individual masses) are very close to their reference value. The same happens for luminosity distance and for the final BH mass and spins.

Despite being compatible with the reference values due to high uncertainties, the predicted values for the spins are rather different from the LIGO-Virgo reference values. Indeed, the measurement of individual spins are difficult due to spin degeneracy of the waveform, as explained in [20] [21] with a study of the PN waveform. As a consequence, only the effective spin parameter $\chi_{eff} = \frac{m_1 s_1 + m_2 s_2}{M}$ can be reliably measured. In fact, the two measures of χ_{eff} agree well with each other: a further confirmation of `mlgw` accuracy. The same feature can be observed on the probability distribution $p(s_1, s_2|\mathcal{D})$ in figure 7: individual spins are poor constrained but the "slope" of their correlation (i.e. the effective spin parameter) is well measured.

We note that our model slightly overestimates the value of the BHs masses. This feature was already noted for a PE run with `TEOBResumS`. The fact that we obtain the same result means that `mlgw` is able to recover with details the training model.

Finally, we see that the two Bayes factors are similar: the Bayesian analysis assigns to both a similar proba-
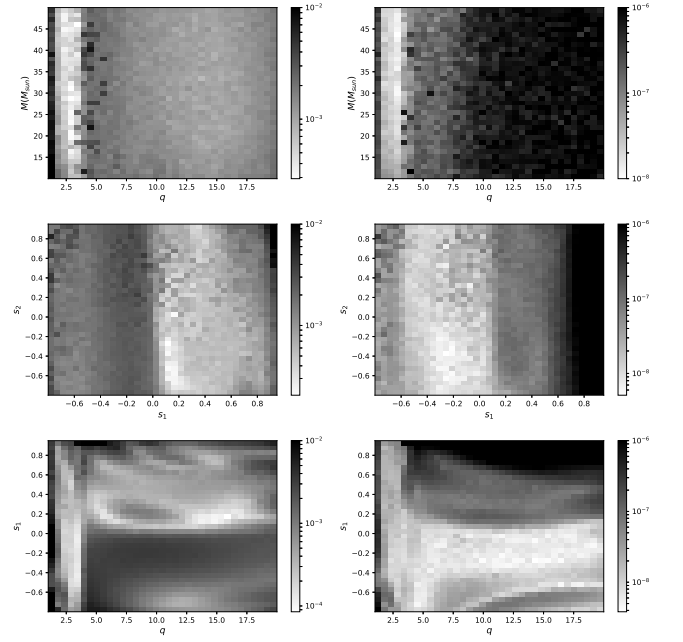

FIG. 9. We report test mismatch and mse for the first PC of the phase, as a function of the orbital parameters $\vartheta = (q, M, s_1, s_2)$. The histograms holds 145061 waveforms, randomly drawn by varying the relevant parameters. In the left column, the color mesh refers to test mismatch; in the right column, we report test mse. On the x-y we display values of two physical quantities, each discretized in 35 bins. On top row, we display q vs M dependence. On central row, we present spins dependence (s_1 vs s_2). On bottom row, we show q vs s_1 dependence. Each wave starts 8 s before merger.

bility that the GW150914 is actually described by the model. However, the evidence for `mlgw` appears to be slightly higher: probably, our priors cover a smaller region than that in the LIGO-Virgo analysis.

C. Runtime analysis

Whean dealing with a real detection scenario, we are often interested in generating a WF which starts from a given frequency f_{min}, which is usually set by the detector's sensitivity window. Thus, it is crucial to measure the speed up that our model can provide in performing such task. We make a comparison with both `TEOBResumS` (the train model and the state-of-the-art in accuracy) and `SEOBNRv4_ROM` (the state-of-the-art in time performance). We define the speed up $\mathcal{S}$ as the ratio between the runtime of benchmark model and the runtime of `mlgw` to produce the a waveform starting from a given f_{min}. Each waveform is produced with constant total mass $M = 100 M_\odot$ and random parameters; the WF is sampled at $f_{sam} = 2048$ Hz. We consider the two cases with $f_{min} = 5$ Hz and $f_{min} = 20$ Hz; the first choice refers to the hypotetical lower bound for the sentivity of the Einstein telescope (ET), while the second is close to
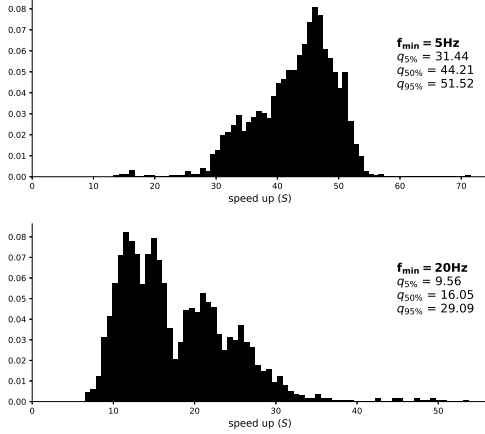
FIG. 10. Histogram for values of the speed up given by `mlgw`, as compared with `TEOBResumS` model, computed on $N = 2000$ test waveforms. Each WF is generated with random physical parameters and has a minimum frequency of 5 Hz (top panel) and 20 Hz (bottom panel). We set a constant total mass $M = 100 \, \mathrm{M_\odot}$ and the sampling rate $f_{sam} = 2048 \, \mathrm{Hz}$. We report the median value $q_{50\%}$ as well as the positions $q_{5\%}$ and $q_{95\%}$ of the 5th and 95th percentile. **Are you really sure that two peaks are fine?? Understand this...**
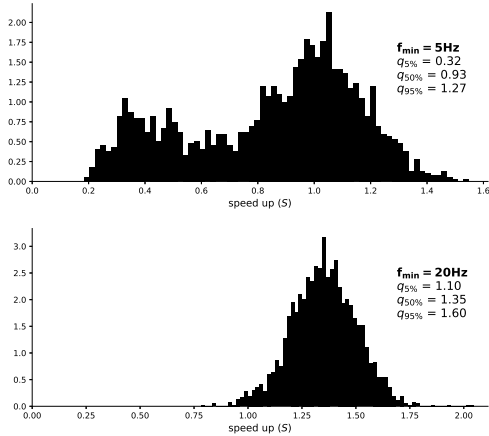

FIG. 11. Histograms for values of the speed up given by `mlgw`, as compared with ROM model `SEOBNRv4_ROM`, computed on $N = 2000$ test waveforms. Each WF is generated with random physical parameters and has a minimum frequency of 5 Hz (top panel) and 20 Hz (bottom panel). We set a constant total mass $M = 100 \, \mathrm{M_\odot}$ and the sampling rate $f_{sam} = 2048 \, \mathrm{Hz}$. We report the median value $q_{50\%}$ as well as the positions $q_{5\%}$ and $q_{95\%}$ of the 5th and 95th percentile.

that of Advanced-LIGO/Virgo.

    a.   Comparison with TEOBResumS In figure 10 we report the histogram of the measured speed up values for model `TEOBResumS`.

    We see that in both cases a substantial speed up is achieved. The speed up is higher for longer WFs [10],

---

[10] This is clearly understood: a longer WF requires more computa-

Task	CPU time (ms)	
	$N_{grid} = 10^3$	$N_{grid} = 10^5$
Generation of raw WF	6.9 (46.9%)	7 (1.6%)
Interpolation to the user grid	4.5 (30.6%)	194 (45.3%)
Post processing	1.7 (11.6%)	206 (48.1%)
Total	14.7 (100%)	428 (100.0%)

TABLE II. Time taken by different stages of the generation of 100 waveforms; data refers to two different values of N_{grid}. Generation of raw WF refers to the computation of the strain $\tilde{h}$ as produced by a ML model. Interpolation to user grid evaluates the WF on the grid chosen by the user. In the post-processing phase, the dependence on d_L, ι and φ_0 is included.

making our model particulary convenient for advanced detectors, with a larger sensitivity window.

    b.   Comparison with SEOBNRv4_ROM In figure 11 we report the histogram of the measured speed up values for model `SEOBNRv4_ROM`. As the ROM model yields WFs in frequency domain, in the time comparison we included a fast Fourier trasform (FFT) of the time domain WF of `mlgw`. This is to ensure that we evaluate the two model at the same conditions. Interestingly, the time taken by the FFT (in the numpy implementation) is similar to that required to generate a WF. Thus for a WF in FD our model cannot be substantially faster, due to the limitation imposed by the FFT.

    We note the the performances are quite similar to each other. The performances of `mlgw` worsen for the 5 Hz case because more grid points are included in the computation, which results in less efficient generation. If a lower sampling rate is employed, `mlgw` outperforms `SEOBNRv4_ROM`.

    However, it is important to stress that `mlgw` is written in pure Python, while `SEOBNRv4_ROM` is coded in C. Thus, the fact the the two models show similar performances is remarkable: surely, a C implementation of `mlgw` would be faster than `SEOBNRv4_ROM`. Furthermore, `mlgw` is more efficient in reconstructing the waveforms: it requires only $O(5 \cdot 10^3)$ WFs for the training, while in order to build a ROM as many as $O(5 \cdot 10^5)$ WFs are required.

    c.   Profiling It is interesting to have a knowledge of the time spent by `mlgw` in each stage of the WF generation procedure. We generate 100 waves with random physical parameters and we measure the CPU time spent to execute each basic task. In table II, we compare the results for two values of N_{grid}.

    We see that the cost of generating the raw WF does not depend on the number of grid points. On the other hand, the interpolation and the post processing depends on N_{grid} and their cost grows dramatically as the user

---

tion for a EOB model, while roughly the same amount of work is done by mlgw

requires more and more points. It is important to stress that the latter two tasks are slow only because they deal with a huge amount of data. Indeed they perform trivial and "quick" operations and their execution relies on well optimized `numpy` routines. If such huge amount of datapoints is required, very little space is left for speed up.

V. FINAL REMARKS AND FUTURE PROSPECTS

We built a ready-to-use Machine Learning model which generates the gravitational wave signal from a binary Black Hole coalescence. The code is released as the package `mlgw` available via the `PyPI` Python package repository. Our model was built for the case of aligned spins (i.e. the non precessing case) and returns a wave given the BH's masses m_1, m_2 and (dimensionless) spins s_1, s_2 taking into account the correct dependence on source luminosity distance d_L, inclination angle ι and reference phase φ_0. Our model shows excellent agreement with the underlying training set. At test times the median mismatch is $\bar{\mathcal{F}} \sim 5 \cdot 10^{-3}$. Lastly, the generation time from `mlgw` is smaller than the underlying training model by a factor of $\sim$????, for a waveform starting at a frequency of $5\,\mathrm{Hz}$. `mlgw` performance matches closely those of a ROM, which is the state-of-the-art for a quick generation.

Remarkably, we discovered that a PCA is able to reproduce a high dimensional wave using a few number of variables. As for the regression model itself, the MoE model, currently it is the "bottleneck" of the model accuracy. For this reason, we explored several alternative regression methods, including neural networks, but none of them showed better performance. Note that underfitting may be an issue whenever the training model shows intrinsic noise due to mis-modeling, e.g. relations that are supposed to be continuous are not, or even due to poor numerical integration schemes.

Our work opens up interesting opportunities in GW data analysis. As shown in fig. 10, the model is the most useful whenever a long waveform is required. Furthermore, the waveform generation time does not depend on the complexity of the underlying surrogate model. The training WFs can be computed with high accuracy, also at high computational cost, without affecting the performance of the WF generation.

This fact is crucial for detection of low frequency signals, as is the case for ET. The analysis of such signals can be performed in the same time required to deal with shorter signals: it will become feasible, even with a small amount of resources, *without any loss of WF quality*. In [] (**plot in email**), it was shown that training model TEOBResumS matches accurately the numerical relativity WFs, even in the case of ET noise curve is used (see plot in the email). Since our model closely reproduces TEOBResumS, it is likely to achieve the target accuracy for the employment in the ET data analysis. **Here I wanted to include the plot sent by Nagar. Is it in a paper? Do we include it here? Or do we just skip it??**

As already noted, the fact that a parameter estimation with `mlgw` is remarkably fast opens up the opportunity for an analysis of the catalog of all the observed GW events. By training (and the training procedure is moderately quick) `mlgw` with different surrogate models, it will be possible to compare their predictions on several observed events. This could allow to detect systematic biases or to prefer a model over another by means of Bayesian model selection (i.e. by comparing different model evidences). A further paper will be devoted to such analysis. **Is there more to say on that?? E.g. about ROM??**

However, our work is far from being over and several issues still require attention.

First of all, the potential speed up of parameter estimation can be even higher, if we take advantage of the closed form expression of the waveform provided by our model. Indeed, a closed form expression for the WF allows for prompt computation of the gradients with respect to the orbital parameters $\boldsymbol{\vartheta} = (m_1, m_2, s_1, s_2)$. Hamiltonian Montecarlo [22] [23], a variant of Markov chain Montecarlo, employs gradients of the waveform to perform an effective sampling of the posterior distibution, which is able to "find quickly" the high density regions. It would provide a state-of-the-art accurate parameter estimation and, at the same time, it will offer a substantial speed up. We believe that this is a promising option and it is among the natural continuations of our work.

Another option, so far never explored, is to use the gradients of the WF for a fast exploration of the likelihood landscape. With any gradient based optimizer, it should be easy to jump to a *local* maximum of the likelihood. Such information can be used to reliably locate a *global* maximum of the likelihood. Once a global maximum is found, the sampling from the posterior distribution should be quicker to perform, yielding an improvement to the overall analysis.

In our model we did not consider precessing system. We made this choice to keep the problem simple. The expertise gained for the simple non precessing case can be applied to deal with the complicacies posed by the precessing case. A good starting point is given in [24]. The dynamics (dependent on 6 parameters) is mapped to a simpler problem, where the precession is controlled by a single parameter. The mapping is as follows:

$$(s_{1x}, s_{1y}, s_{1z}) \longmapsto (s_P, 0, s_{1z})$$
$$(s_{2x}, s_{2y}, s_{2z}) \longmapsto (0, 0, s_{2z})$$

where $s_P = s_P(q, s_{1x}, s_{1y}, s_{2x}, s_{2y})$ (see [24, eq. 3]). In the authors' word, the effective spin parameter s_P "accurately captures the dominant precession-induced features in GW signals across the full parameter space". The waveform dependence on the effective spin parameter s_P can be fitted by a ML model in the same fashion of the

other orbital parameters, thus simplyfing the regression problem.

All surrogate models provides numerical solutions to approximate form of the Einstein equations. They are useful to catch the dominant physics but we expect them to show a degree of inaccuracy, especially close to coalescence. Numerical relativity solves numerically the unapproximated Einstein equations and so far yields the best available solutions. Clearly, NR waveforms are much slow to generate and cannot be used in parameter estimation. Our model could be trained on the publicly available NR waveforms catalogs (see e.g. [25] and [26]) and would provide the best generalization of the numerical waveform. If the model proved to be reliable and effective, we could dispense with the surrogate models altogether and use only NR inputs for parameter estimation. This would ensure that all the relevant physics close to merger is adequately captured. Furthermore, a training set of NR waveforms can be used to address problems where an approximation to Einstein equations (such as PN expansion) is not available.

Are you sure of it??? Probably you should say that you don't know enough on the matter and that you will have to investigate...

Unfortunately, at the moment there are too few NR waveforms ($O(10^2)$) available to perform a reliable training: we need at least $O(10^3)$ waveforms. However, in the future we expect more and more waveforms to be included in the dataset, eventually allowing for a reliable training. Furthermore, NR waveforms are too short as compared to detected signals in interferometers. Probably, some form of hybridization from PN models is required to tackle the inspiral phase.

Our machine learning approach to signal generation is very flexible and it provides a general framework. We expect it to work for every source for which a number of training waveforms are available (also those for which no surrogate models are available). Machine learning models for various range of sources can be crucial in the future, where signals from a number of different sources are expected to be detected. In that scenario, a parameter estimation must be able to detect among different source and this will require a lot of computational work. Speed up will be more pressing.

[1] J. Aasi *et al.*, "Parameter estimation for compact binary coalescence signals with the first generation gravitational-wave detector network," *Phys. Rev.*, vol. D88, p. 062001, 2013.

[2] J. Veitch *et al.*, "Parameter estimation for compact binaries with ground-based gravitational-wave observations using the LALInference software library," *Phys. Rev.*, vol. D91, no. 4, p. 042003, 2015.

[3] A. Buonanno and T. Damour, "Effective one-body approach to general relativistic two-body dynamics," *Phys. Rev. D*, vol. 59, p. 084006, Mar 1999.

[4] T. Damour and A. Nagar, "The Effective One Body description of the Two-Body problem," *arXiv e-prints*, p. arXiv:0906.1769, Jun 2009.

[5] A. Nagar, S. Bernuzzi, W. Del Pozzo, G. Riemenschneider, S. Akcay, G. Carullo, P. Fleig, S. Babak, K. W. Tsang, M. Colleoni, and et al., "Time-domain effective-one-body gravitational waveforms for coalescing compact binaries with nonprecessing spins, tides, and self-spin effects," *Physical Review D*, vol. 98, Nov 2018.

[6] M. Pürrer, "Frequency-domain reduced order models for gravitational waves from aligned-spin compact binaries," *Classical and Quantum Gravity*, vol. 31, p. 195010, Sep 2014.

[7] M. Pürrer, "Frequency domain reduced order model of aligned-spin effective-one-body waveforms with generic mass ratios and spins," *Physical Review D*, vol. 93, Mar 2016.

[8] A. Bohé, L. Shao, A. Taracchini, A. Buonanno, S. Babak, I. W. Harry, I. Hinder, S. Ossokine, M. Pürrer, V. Raymond, and et al., "Improved effective-one-body model of spinning, nonprecessing binary black holes for the era of gravitational-wave astrophysics with advanced detectors," *Physical Review D*, vol. 95, Feb 2017.

[9] P. Ajith, M. Boyle, D. A. Brown, S. Fairhurst, M. Hannam, I. Hinder, S. Husa, B. Krishnan, R. A. Mercer, F. Ohme, C. D. Ott, J. S. Read, L. Santamaria, and J. T. Whelan, "Data formats for numerical relativity waves," 2007.

[10] A. Arbey and J.-F. Coupechoux, "Black hole mergers, gravitational waves and scaling relations," 2019.

[11] K. Murphy, *Machine Learning: A Probabilistic Perspective.* Adaptive Computation and Machine Learning series, MIT Press, 2012.

[12] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, pp. 79–87, 1991.

[13] F. Ohme, A. B. Nielsen, D. Keppel, and A. Lundgren, "Statistical and systematic errors for gravitational-wave inspiral signals: A principal component analysis," *Physical Review D*, vol. 88, Aug 2013.

[14] A. et al., "Observation of gravitational waves from a binary black hole merger," *Physical Review Letters*, vol. 116, Feb 2016.

[15] J. Skilling, "Nested sampling for general bayesian computation," *Bayesian Analysis*, vol. 1, pp. 833–860, 12 2006.

[16] B. Abbott, R. Abbott, T. Abbott, M. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. Adhikari, and et al., "Properties of the binary black hole merger gw150914," *Physical Review Letters*, vol. 116, Jun 2016.

[17] B. Abbott, R. Abbott, T. Abbott, M. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. Adhikari, and et al., "Properties of the binary black hole merger gw150914," *Physical Review Letters*, vol. 116, Jun 2016.

[18] X. Jiménez-Forteza, D. Keitel, S. Husa, M. Hannam, S. Khan, and M. Pürrer, "Hierarchical data-driven approach to fitting numerical relativity data for nonprecessing binary black holes with an application to final spin

and radiated energy," *Phys. Rev.*, vol. D95, no. 6, p. 064024, 2017.

[19] D. Foreman-Mackey, "corner.py: Scatterplot matrices in python," *The Journal of Open Source Software*, vol. 24, 2016.

[20] E. Baird, S. Fairhurst, M. Hannam, and P. Murphy, "Degeneracy between mass and spin in black-hole-binary waveforms," *Phys. Rev. D*, vol. 87, p. 024035, Jan. 2013.

[21] M. Pürrer, M. Hannam, and F. Ohme, "Can we measure individual black-hole spins from gravitational-wave observations?," *Phys. Rev.*, vol. D93, no. 8, p. 084042, 2016.

[22] M. Betancourt, "A conceptual introduction to hamiltonian monte carlo," 2017.

[23] E. K. Porter and J. Carré, "A hamiltonian monte–carlo method for bayesian inference of supermassive black hole binaries," *Classical and Quantum Gravity*, vol. 31, p. 145004, Jul 2014.

[24] P. Schmidt, F. Ohme, and M. Hannam, "Towards models of gravitational waveforms from generic binaries: Modelling precession effects with a single effective precession parameter," *Physical Review D*, vol. 91, Jan 2015.

[25] A. H. Mroue *et al.*, "Catalog of 174 Binary Black Hole Simulations for Gravitational Wave Astronomy," *Phys. Rev. Lett.*, vol. 111, no. 24, p. 241104, 2013.

[26] J. Healy, C. O. Lousto, Y. Zlochower, and M. Campanelli, "The rit binary black hole simulations catalog," *Classical and Quantum Gravity*, vol. 34, p. 224001, Oct 2017.