Realizing the potential of deep neural network for analyzing neutron star observables and dense matter equation of state

Ameya Thete[*] and Kinjal Banerjee[†]
Department of Physics, BITS-Pilani, K. K. Birla Goa Campus, Goa 403726, India

Tuhin Malik[‡]
CFisUC, Department of Physics, University of Coimbra, P-3004–516 Coimbra, Portugal

(Received 3 August 2023; accepted 1 September 2023; published 26 September 2023)

The difficulty in describing the equation of state (EOS) for nuclear matter at densities above the saturation density (ρ_0) has led to the emergence of a multitude of models based on different assumptions and techniques. These equations of state, when used to describe a neutron star (NS), lead to differing values of observables. An outstanding goal in astrophysics is to constrain the dense matter EOS by exploiting astrophysical and gravitational wave measurements. Nuclear matter parameters appear as Taylor coefficients in the expansion of the EOS around the saturation density of symmetric and asymmetric nuclear matter and provide a physically motivated representation of the EOS. In this paper, we introduce a deep learning-based methodology to predict key neutron stars observables such as the NS mass, NS radius, and tidal deformability from a set of nuclear matter parameters. Using generated mock data, we confirm that the neural network model is able to accurately capture the underlying physics of finite nuclei and replicate intercorrelations between the symmetry energy slope, its curvature, and the tidal deformability arising from a set of physical constraints. We also test our network with mock data generated by a different class of physics model, which was not part of the training, to explore the limitations of model dependency in the results. We also study the validity of our trained model using Bayesian inference and show that the performance of our model is on par with physics-based models with the added benefit of much lower computational cost.

DOI: 10.1103/PhysRevD.108.063028

I. INTRODUCTION

Neutron stars (NSs) are some of the most fascinating astrophysical objects in multimessenger astronomy. They contain matter at extreme conditions far beyond the ones accessible in a terrestrial laboratory. The core of a NS is believed to contain matter at a few times nuclear saturation density, ρ_0 [1–4].[1] While the structure of a NS can be determined using the Tolman-Oppenheimer-Volkoff (TOV) equations [5,6], this requires the knowledge of the dense matter equation of state (EOS). Understanding the internal structure of neutron stars in terms of fundamental interactions between its constituents is an open problem in nuclear physics. To understand the physics of dense matter, at low densities of $\rho \sim 1$–$2\rho_0$, we can use *ab initio* approaches derived from chiral effective field theory [7–11]. At large densities of $\rho \geq 40\rho_0$, perturbative quantum chromodynamics (QCD) calculations converge and provide reliable estimates [11–14]. However, in the intermediate-density region around $\rho \sim 2$–$10\rho_0$ which is relevant for most structural descriptions of neutron stars, reliable calculations from first principles are currently unavailable [11,15]. Quantum field theory calculations based on lattice QCD are challenging at these densities due to the sign problem that arises in Monte Carlo simulations [16]. As a result, structural descriptions of neutron stars rely on relativistic and nonrelativistic phenomenological models for the EOS. The nuclear matter parameters (NMPs), which form the basis of the construction of these equations of state for neutron star matter, are not directly accessible. While lower-order NMPs can be empirically extracted through finite nuclei nuclear physics experiments [17–20], in order to constrain higher-order NMPs, we need to rely on astrophysical observations [21–23].

Recent developments in multimessenger astronomy have provided important information about high-density nuclear matter physics relevant for NSs. Constraining the EOS is a joint task between nuclear physics and astrophysics. Measured astrophysical quantities such as NS observables can uncover properties of dense nuclear matter. Narrowing

[*] f20180885@goa.bits-pilani.ac.in
[†] kinjalb@gmail.co
[‡] tuhin.malik@uc.pt
[1] $\rho_0 = 0.16\ \mathrm{fm}^{-3}$.

the constraints on NS observables, therefore, has the ability to constrain the behavior of matter under extreme conditions. It is expected that precise and simultaneous measurements of NS properties like mass, radius, moment of inertia, and tidal deformability may help constrain the EOS to a narrow range [24–27]. High-mass pulsars like PSR J1614–2230 ($M = 1.908 \pm 0.016 M_\odot$) [28–30], PSR J0348–0432 ($M = 2.01 \pm 0.04 M_\odot$) [31], PSR J0740 + 6620 ($M = 2.07 \pm 0.07 M_\odot$ [32], and very recently PSR J1810 + 1714 ($M = 2.13 \pm 0.04 M_\odot$) [33] have already placed tight constraints on the EOS. Gravitational wave (GW) signals emitted from a NS merger event depend on the behavior of the neutron star matter at high densities [34,35]. Therefore, the values of tidal deformability obtained from GW events such as GW170817 associated with a binary NS merger, as well as the simultaneous measurement of NS masses and radii from high-precision x-ray space missions, such as Neutron Star Interior Composition Explorer (NICER), may help further constrain the EOS. Some current observational evidences are the simultaneous measurements of NS mass $1.34^{+0.15}_{-0.16} M_\odot$ and radius $12.71^{+1.14}_{-1.19}$ km for the pulsar PSR J0030 + 0451 by NICER [36]. Other independent analyses show that the radius is $13.02^{+1.24}_{-1.06}$ km and the mass is $1.44^{+0.15}_{-0.14} M_\odot$ [37]. However, the recent measurement of the equatorial circumferential radius of the pulsar PSR J0740 + 6620 with mass $M = 2.072^{+0.067}_{-0.066} M_\odot$ and $R = 12.39^{+1.30}_{-0.98}$ km [68% confidence interval (CI)] [38] cannot further constrain the EOS which already predicts a NS with maximum mass more than $2M_\odot$ [39].

The EOS—to a good approximation—can be expressed in terms of NMPs at saturation density. The NMPs usually considered for constructing the EOS are the incompressibility coefficient, the skewness parameter of the symmetric nuclear matter, the symmetry energy coefficient, its slope, and the curvature parameters characterizing the density dependence of the symmetry energy. Recently, there has been a comprehensive analysis of correlations of tidal deformability and other NS properties with NMPs [21,40,41]; however, these correlations are found to be model dependent [41,42]. Reference [43] shows that the correlations are sensitive to finite nuclei properties, which are accessible to laboratories. Therefore, finite nuclei properties are important quantities to consider in a model while determining the correlations between NS properties and NMPs. Of late, the equations of state obtained by several metamodels have gained popularity owing to their cost effectiveness in big simulations [13,24,44]. These models are constrained by *ab initio* theoretical calculations of nucleon-nucleon chiral potentials for low-density neutrons and nuclear matter [45,46] and perturbative QCD for asymptotically high-density regimes [13]. In the intermediate-density region, the EOS is evolved in a thermodynamically consistent manner with either piecewise polytropic segments [47–49], a speed-of-sound interpolation, or a spectral interpolation [50,51]. These metamodels are limited to incorporating finite nuclei properties and differ on results in establishing a bridge between NS properties and NMPs. Nonparametric models of the NS EOS have also been proposed based on Gaussian processes [52,53] which use Bayesian methods to infer the EOS from multimessenger data. These models are usually computationally expensive to implement or are highly sensitive to the choice of the training datasets and, therefore, might be limited by the current knowledge of the EOS [54]. Consequently, there is a need to search for alternative approaches to construct a model-independent EOS.

In recent years, deep learning (DL) has been extensively applied to a wide range of technological and scientific tasks. DL algorithms, which are a class of machine learning (ML) algorithms, are highly scalable and distributed computational techniques with the ability to learn intricate relationships from raw data using units called neurons arranged in a stacked fashion. The advent of high-performance computing and the development of parallel devices like graphics processing units have rendered DL as the primary choice of algorithms for tasks such as computer vision [55] or natural language processing [56]. DL models have been used as alternatives to conventional statistical framework and have been successfully applied to many problems in physics. Most applications of ML and DL in physics have been in analyzing data obtained from data-intensive experiments like LIGO for the detection and denoising of GW signals [57–59] and the Large Hadron Collider for particle track reconstruction or anomaly detection [60–62]. Significant progress has also been made in using these algorithms in the context of nuclear physics [63] and neutron star physics [54,64,65]. While these applications are certainly promising, it remains to be seen up to what extent a DL model can supplement existing physical models. Recent research also aims to address whether a trained DL model can produce correct predictions from experimental data alone and whether such predictions are comparable to mesoscopic phenomenological physics models [66]. Such ML- and DL-based models do not have the feature richness possessed by physics-based models, but they offer other benefits like cost effectiveness when dealing with a large amount of experimental or observational data. Recent works [15,64,67] have studied the applications of machine learning methods to the neutron star EOS. They employ a feed-forward neural network (FFNN) to map neutron star data to EOS parameters. Instead of considering the FFNN as merely an interpolation tool between EOS NMPs and neutron star observables, we adopt an approach similar to [54] by treating the FFNN as a representation of the EOS itself.

In this paper, our focus is on implementing an artificial neural network to emulate a physics model and assess its accuracy through validation and inference tasks. Our primary goal is to thoroughly examine the neural network's ability to mimic the physics model and subsequently utilize the trained network for computationally expensive Bayesian inference tasks. To the best of our knowledge, the use of neural networks in a Bayesian inference framework has not been employed in nuclear astrophysics literature. We show that we can indeed obtain results comparable to physics models at a fraction of the computational cost. This points out a novel utilization of machine learning and neural network models in research. Let us however clarify that we are not claiming that neural networks can replace physics-based models. Our claim is that such models can help us capture some of the features of the EOS at much lower computational costs.

The paper is structured as follows: in Sec. II, we discuss the parametrization of the EOS which leads to the emergence of NMPs, followed by a brief review of the theory of artificial neural networks and the DL approach used to map NMPs to NS observables. This is followed by a description of the Bayesian statistical framework. In Sec. III, we test how well the best fit neural network model captures correlations between EOS parameters. We also perform a Bayesian analysis to compare our models with a physics-based model and show that the computational cost is much less for our model. We finally end with our conclusions in Sec. IV.

II. FRAMEWORK

In this section, we outline the different facets of the adopted framework for our analysis. In Sec. II A, we briefly describe the neutron star EOS and the nuclear matter parameters involved in its construction. Then, in Sec. II B, we describe the generation of the dataset used to train the neural network model. Section II C describes at length the DL approach used to construct a deep neural network model which accepts nuclear matter saturation parameters as inputs and produces neutron star properties obtained from a set of realistic nuclear physics equations of state as targets. Finally, in Sec. II D, we present a Bayesian inference framework that is applied on the trained neural network model to verify the performance of our model in comparison with Skyrme models.

A. Nuclear matter parameters

The structure of neutron stars is obtained by solving the TOV equation with a given EOS for the nuclear matter. The EOS is expressed as the variation of pressure e with density ρ, over a wide range of densities. To a good approximation, any EOS calculated from phenomenological nuclear models can be decomposed into two parts: (i) the EOS for symmetric nuclear matter $e(\rho, 0)$ and (ii) a term involving the symmetry energy coefficient $S(\rho)$ and the asymmetry δ,

$$e(\rho, \delta) \simeq e(\rho, 0) + S(\rho)\delta^2, \tag{1}$$

where $\rho = \rho_n + \rho_p$ is the baryon density, ρ_n and ρ_p are the neutron and proton densities, respectively, and the asymmetry $\delta = (\rho_n - \rho_p)/\rho$. We can then characterize the density dependence of the energy density of symmetric matter around the saturation density ρ_0 in terms of a few bulk parameters by constructing a Taylor expansion around ρ_0. That is,

$$e(\rho, 0) = e_0 + \frac{K_0}{2}\left(\frac{\rho - \rho_0}{3\rho_0}\right)^2 + \frac{Q_0}{6}\left(\frac{\rho - \rho_0}{3\rho_0}\right)^3 + \mathcal{O}(4). \tag{2}$$

The coefficients e_0, K_0, and Q_0 denote the energy per particle, the incompressibility coefficient, and the third derivative of symmetric matter at saturation density, respectively. Similarly, the behavior of the symmetry energy around saturation can also be characterized in terms of a few bulk parameters,

$$S(\rho) = J_{\mathrm{sym},0} + L_{\mathrm{sym},0}\left(\frac{\rho - \rho_0}{3\rho_0}\right) + \frac{K_{\mathrm{sym},0}}{2}\left(\frac{\rho - \rho_0}{3\rho_0}\right)^2 + \mathcal{O}(3), \tag{3}$$

where $J_{\mathrm{sym},0} = S(\rho_0)$ is the symmetry energy at saturation density. The incompressibility K_0, the skewness coefficient Q_0, the symmetry energy slope $L_{\mathrm{sym},0}$, and its curvature $K_{\mathrm{sym},0}$ evaluated at saturation density are defined in [68]. These quantities are the key NMPs that describe any EOS. Hence, an EOS can be represented by a point in the seven-dimensional parameter space of NMPs $\{e_0, \rho_0, K_0, Q_0, J_{\mathrm{sym},0}, L_{\mathrm{sym},0}, \text{and} K_{\mathrm{sym},0}\}$ [43]. Symbolically, the jth EOS in this space is written as

$$\begin{aligned} \mathrm{EOS}_j &= \{e_0, \rho_0, K_0, Q_0, J_{\mathrm{sym},0}, L_{\mathrm{sym},0}, \text{and}\, K_{\mathrm{sym},0}\}_j \\ &\approx \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \end{aligned} \tag{4}$$

where $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate Gaussian distribution with $\boldsymbol{\mu}$ being the mean value of the nuclear matter parameters $\mathbf{p}$ and a covariance matrix $\boldsymbol{\Sigma}$. The diagonal elements of $\boldsymbol{\Sigma}$ represent the variance or the squared error for the parameters p_i. The off-diagonal elements of $\boldsymbol{\Sigma}$ are the covariances between different parameters p_i and p_j, and denote the correlation coefficient between them. Hence, given a mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, a large number of equations of state can be obtained.

While the Taylor expansion of the symmetric and asymmetric energy is only truly accurate around the saturation density, we can treat these expansions as a parametrization of the EOS—similar to other adopted parametrizations—with the condition that this representation asymptotically approaches the Taylor expansion in the limit $\rho \to \rho_0$ [21,69]. This lets us ignore any issue arising with the convergence of the approximation. However, the higher-order NMPs obtained via this parametrization might markedly diverge from the actual nuclear matter expansion coefficients. They can be thought of as effective parameters that incorporate the effects of missing higher-order terms. Moreover, Refs. [26,69] indicate that an EOS obtained from the Skyrme framework can be well reproduced by considering Taylor coefficients until the third or fourth order.

B. Dataset generation

In our analysis, the input data used to train an artificial neural network are the seven key nuclear matter parameters that govern the equation of state $\{e_0, \rho_0, K_0, Q_0, J_{\mathrm{sym},0}, L_{\mathrm{sym},0}, K_{\mathrm{sym},0}\}$. Six neutron star properties are the target variables: the maximum NS mass $M_{\max}$, the maximum NS radius $R_{\max}$, the radius for $1.4M_\odot$ NS $R_{1.4}$, and the tidal deformability Λ_M for NS having mass $M \in [1.0, 1.4, 1.8]M_\odot$. We generate our dataset by sampling points from the multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ is the mean vector with components $\mu_i = \mathbb{E}[p_i]$ and $\boldsymbol{\Sigma}$ is the covariance matrix with entries $\Sigma_{ij} = \mathbb{E}[(p_i - \mu_i)(p_j - \mu_j)]$, for a NMP p.[2] This method closely follows the procedure for generating the Case-II dataset in Ref. [43]. We assume an *a priori* intercorrelation coefficient between $L_{\mathrm{sym},0}$ and $K_{\mathrm{sym},0}$ of 0.8, which is a reasonable choice for nuclear physics models that satisfy finite nuclear properties. Models that satisfy finite nuclear properties exhibit different correlations between NMPs and NS properties as compared to metamodels or nuclear physics models for infinite nuclear matter that do not respect finite nuclear properties [41,42]. Therefore, we consider Case-II data of Ref. [43] to mimic the microphysics information of finite nuclear properties. Figure 1 presents the 7×7 matrix for the correlation coefficients between the NMPs in the sampled data. The central values and uncertainties on each NMP in the constructed distribution are listed in Table I.

We then construct the Skyrme EOS from a drawn NMP sample and check whether the EOS (a) predicts a NS maximum mass above $2M_\odot$, (b) predicts a tidal deformability for a NS with $M = 1.4M_\odot$ below 800, and (c) satisfies the causality condition, i.e., the speed of sound $c_s = \sqrt{\frac{\mathrm{d}p}{\mathrm{d}e}} \leq c$ at the center of maximum mass NS, where c is the speed of light in vacuum. Any samples that generate equations of state that do not satisfy these conditions are discarded. For each EOS, the TOV and deformability equations [70] are solved to obtain the six aforementioned NS observables. Naturally, realistic NS observations accrue experimental and instrumental errors, which result in corresponding uncertainties while reconstructing the NMPs. For simplicity, we choose to disregard NS observational errors and uncertainties while training the model. Following this procedure, we obtain 2106 filtered NMPs and corresponding NS observables which form the dataset.

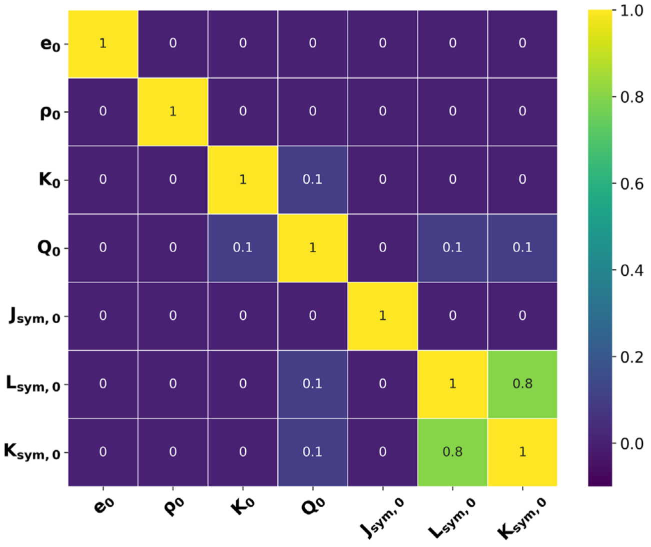

FIG. 1. Correlations among the various NMPs for the sampled dataset. Correlations among the off-diagonal pairs corresponding to $L_{\mathrm{sym},0} - K_{\mathrm{sym},0}$ are the only nontrivial values. The figure displays sample statistics that reveal insignificant correlations between other NMPs.

C. Artificial neural networks

Artificial neural networks (ANNs) are a class of machine learning algorithms that have found widespread use over the past decade. The popularity of ANNs arises from their ability to model complex nonlinear relationships within data and their potential to generalize to a wide class of functions. An ANN with sufficiently many layers or neurons is theoretically capable of representing any continuous function [71,72]. Therefore, as a general rule,

TABLE I. The mean value $\boldsymbol{\mu}_{\boldsymbol{p}_i}$ and error $\sqrt{\boldsymbol{\Sigma}_{\boldsymbol{p}_i\boldsymbol{p}_i}}$ for the nuclear matter parameters p_i employed for the multivariate Gaussian distribution (MVGD). All quantities are in units of MeV except for ρ_0 which is in units of fm^{-3}.

NMP	MVGD		Training Set	
$\boldsymbol{p_i}$	$\boldsymbol{\mu_{p_i}}$	$\sqrt{\boldsymbol{\Sigma_{p_ip_i}}}$	$\boldsymbol{\mu_{p_i}}$	$\sqrt{\boldsymbol{\Sigma_{p_ip_i}}}$
e_0	−16.0	0.25	−15.99	0.25
ρ_0	0.16	0.005	0.158	0.005
K_0	230	20	242	15
Q_0	−300	100	−307	70
$J_{\mathrm{sym},0}$	32	3	32.65	2.8
$L_{\mathrm{sym},0}$	60	20	73	12
$K_{\mathrm{sym},0}$	−100	100	−20	50

[2] $\mathbb{E}[X]$ is the expectation value of a random variable X and is defined as $\mathbb{E}[X] = \int_{-\infty}^{\infty} xp(x)$, where $p(x)$ is the probability density function.

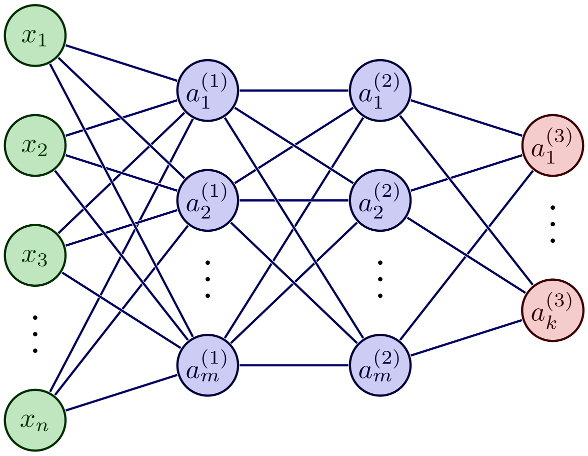

FIG. 2. An example of a fully connected feed-forward neural network with two hidden layers. Each hidden layer consists of m neurons, denoted by $a_j^{(l)}$, where l denotes an index over the hidden layers and the index j runs over the dimension of the hidden layer. This particular network accepts an n-dimensional input through the input layer (in green), passes the input through the hidden layers (in lavender), and produces a k-dimensional output at the output layer (in red). Solid lines between individual nodes denote weighted connections that are learned during the training procedure. Activation functions are not shown in this figure.

a more complex network with a larger number of parameters is able to learn more abstract features from the data.

Feed-forward neural networks are the simplest type of neural network architecture. For an input x, the objective of a FFNN is to approximate some true mapping $f^*(\mathrm{x})$ by a mapping $f(\mathrm{x};\boldsymbol{\theta})$, parametrized by a set of weights $\boldsymbol{\theta}$. A typical feed-forward neural network consists of a number of processing units called neurons, arranged into one or many layers composed in a sequential fashion. A neuron performs a linear operation by aggregating weighted inputs received from neurons in the previous layer. A FFNN generally consists of an input layer, followed by one or more hidden layers, and a final output layer consisting of one or more neurons. Computation in a FFNN flows in a linear fashion, starting at the input layer and moving successively through the hidden layers until it reaches the output layer. Figure 2 provides an illustration of a simple feed-forward neural network with one input layer, two hidden layers, and an output layer.

The parameters $\boldsymbol{\theta}$ typically represent the weights assigned to a connection between neurons in adjacent layers. The training data provide noisy approximate examples of $f^*(\mathrm{x})$ and each x is paired with a corresponding label y. The goal of a learning algorithm is to learn a particular value of $\boldsymbol{\theta}$ that results in the best function approximation. During the training procedure, the learning algorithm does not say what each layer does but instead decides how to use these layers to produce an optimal approximation of f^*. As the training data do not show the desired output for intermediate layers, they are called hidden layers. The number of hidden layers decides the depth of the network, and the dimensionality or the number of neurons in each hidden layer determines the width of the network. To introduce nonlinearity in the computation between two successive layers within the model, a nonlinear function, called the activation function, acts elementwise on the output of one hidden layer, and the output of the function is passed to the next layer in the computation. For most modern neural networks, the default recommendation is to use the rectified linear unit (ReLU) [73,74] activation, defined as $g(z) = \max\{0, z\}$. Other commonly used choices for the activation function include the tanh function or the logistic function. For a textbook review of neural networks and training algorithms, see Ref. [75].

The ability of neural networks to model highly nonlinear relationships between input and output variables makes them ideal for estimating neutron star properties from the equation of state parameters. This is important because the relationships between these two quantities are expected to be nontrivial and involve multiple intermediate steps composed of nonlinear operations. Moreover, a neural network offers two major advantages over a conventional approach with traditional physics models:

(1) An ANN can efficiently map the sample NMPs to NS properties without calculating the EOS from nuclear physics models. Similar works [76,77] demonstrate that ANNs offer up to a twofold speedup over conventional astrophysical models.

(2) ANNs can also accurately capture finite nuclear information, which can be computationally expensive to verify with a traditional physics model in a Bayesian setting.

At this step, we wish to note that other machine learning models can also be used for an identical purpose, albeit with varying degrees of success. We performed a preliminary comparison of FFNNs with other ML models, specifically linear regression, support vector regression, and XGBoost (eXtreme Gradient Boosting) [78]. We observed that FFNNs outperformed all the other models in the study, and therefore, we chose to use FFNNs for this work.

In supervised learning instances, the dataset is generally partitioned into nonoverlapping training, testing, and validation sets. The training dataset is used by the neural network to learn, the validation set is used to see whether the network is learning properly during the training procedure and to select the optimal values for the hyperparameters, and the testing dataset is used to assess the performance of the trained model. It should be noted that the model does not see any instances from the testing dataset until after the training is completed. Accordingly, we randomly partition the original generated dataset following a 70%-20%-10% split into a training set, a validation set, and a testing set consisting of 1515, 422, and 169 instances, respectively. Before training, the features in the split datasets are standardized by removing the mean and scaling them to unit variance. Standardization of data is a common requirement for many ML algorithms as they tend to perform poorly if the individual features are not standard normally distributed, i.e., Gaussian with unit variance and zero means. Standardization is performed using Scikit-Learn's [79] StandardScaler method. The output of the network is scaled back to the original distribution before computing the loss.

The choice of the ANN architecture, such as the number of layers and the number of neurons in each layer, represents a trade-off between the quality of fit and the overfitting problem. The model should contain sufficiently many trainable parameters to learn the ground truth function accurately. At the same time, the ANN must avoid overfitting the training data by losing its ability to generalize well. As there is no panacea for choosing an optimal model configuration, the final architecture for the ANN was chosen based on empirical tests performed on the data and selecting the set of configurations that performed best on the validation set. Our network consists of two hidden layers with 15 neurons each and uses the ReLU activation function, which is a standard choice for most deep learning applications and is known to mitigate the vanishing gradients problem [80]. The ReLU activation function behaves as an identity function for positive inputs and saturates at 0 for negative inputs. No activation function is applied to the input or the output layer. The design of our feed-forward architecture is summarized in Table II.

The neural network is implemented and trained using the PYTHON Keras library [81] with a TensorFlow [82] backend. Neural network parameters are initialized with the Glorot uniform distribution [83]. We use the Adam optimizer [84] to update the weights of the ANN. For training, we use an initial learning rate of 1×10^{-3} and the training data are batched into batches of size 16. Training is performed for a total of 40 epochs or until the validation loss stops decreasing. The model is trained on a 2-core Intel Xeon CPU @ 2.20 GHz. The network seeks to optimize a root-mean-square error (RMSE) loss between the predictions and actual neutron star properties.

TABLE II. Neural network architecture used in this work. The input layer consists of seven neurons corresponding to nuclear matter parameters. In the output layer, the six neurons correspond to the six neutron star properties.

Layer	Number of neurons	Activation function
0 (input)	7	None
1, 2	15	ReLU
3 (output)	6	None

D. Bayesian statistical framework

The Bayesian statistical framework allows us to carry out a detailed analysis of the parameters of a model for a given set of fit data [85–87]. The hypothesis or the prior knowledge of the model parameters and various constraints on them is encoded through the prior distributions. The technique yields a joint posterior distribution of the model parameters by updating the probability for the hypothesis using the available observational data according to the Bayes theorem. The posterior distribution of the model parameters $\boldsymbol{\theta}$ can be written as

$$P(\boldsymbol{\theta}|D) = \frac{\mathcal{L}(D|\boldsymbol{\theta})P(\boldsymbol{\theta})}{\mathcal{Z}}, \tag{5}$$

where $\boldsymbol{\theta}$ and D denote the set of model parameters and the fit data, respectively. $P(\boldsymbol{\theta}|D)$ is the joint posterior probability of the parameters, $\mathcal{L}(D|\boldsymbol{\theta})$ is the likelihood function, $P(\boldsymbol{\theta})$ is the prior for the model parameters, and $\mathcal{Z}$ is the evidence. The posterior distribution of a given parameter an be obtained by marginalizing $P(\boldsymbol{\theta}|D)$ over remaining parameters. The marginalized posterior distribution for a parameter θ_i is obtained as

$$P(\theta_i|D) = \int P(\boldsymbol{\theta}|D)\prod_{j\neq i} d\theta_j. \tag{6}$$

A major contribution of this study is to demonstrate how a neural network can be effectively integrated within a Bayesian inference framework for astrophysical use cases. In particular, we demonstrate the effect of intercorrelations between symmetry energy parameters on NS properties. We sample NMPs within a Bayesian framework in two different ways: (i) sampling all the NMPs randomly without any intercorrelations among them and (ii) sampling all the NMPs randomly, except for the pair $L_{\mathrm{sym},0}$-$K_{\mathrm{sym},0}$ which is set to a correlation coefficient of 0.9. Then, we compare the posterior distribution of the NMPs for both cases under certain constraints on NS properties.

For the sampling process, we use the Gaussian likelihood function defined as

$$\mathcal{L}(D|\boldsymbol{\theta}) = \prod_j \left(\frac{1}{\sqrt{2\pi}\sigma_j}\right) \exp\left[-\frac{1}{2}\sum_{i=1}^{N_d}\left(\frac{d_i - m_j(\boldsymbol{\theta})}{\sigma_j}\right)^2\right]. \tag{7}$$

Here the index j runs over all the data, d_j and m_j are the data and the corresponding model values, respectively, the model m is parametrized by a set of parameters $\boldsymbol{\theta}$, and σ_j are the adopted uncertainties. The evidence is used to compare the compatibility of different models with the available data. In our present work, the evidence $\mathcal{Z}$ is not relevant and thus can be ignored. To obtain the marginalized posterior distributions of the NMPs within the

TABLE III. The mean value $\boldsymbol{\mu}_{p_i}$ and error $\sqrt{\boldsymbol{\Sigma}_{p_i p_i}}$ for the nuclear matter parameters p_i in the prior multivariate Gaussian distribution. All quantities are in units of MeV except for ρ_0 which is in units of fm^{-3}.

NMP	P1		P2	
p_i	μ_{p_i}	$\sqrt{\Sigma_{p_i p_i}}$	μ_{p_i}	$\sqrt{\Sigma_{p_i p_i}}$
e_0	−15.99	0.25	−15.99	0.50
ρ_0	0.158	0.005	0.158	0.01
K_0	242	15	242	30
Q_0	−307	70	−307	140
$J_{\mathrm{sym},0}$	32.65	2.8	32.65	5.6
$L_{\mathrm{sym},0}$	73	12	73	24
$K_{\mathrm{sym},0}$	−20	50	−20	100

Bayesian framework, we require a set of fit data, a model, and a set of priors for the nuclear matter parameters. The likelihood function for a given set of fit data is evaluated for a sample of NMPs populated according to their prior distributions. The joint probability distribution of the NMPs is obtained using the product of the likelihood function and the prior distribution. The fit data for the likelihood function is provided by the neural network for a sample drawn from the prior distribution. To compute the likelihood, we use the maximum neutron star mass M_{max}, the maximum radius R_{max}, the radius $R_{1.4}$, and the tidal deformability $\Lambda_{1.4}$ from the set of outputs generated by the neural network for a given input of NMPs. Instead of using a distinct value for each data point, we fix d_i in Eq. (7) to a mean value μ_j. Therefore, for our study, Eq. (7) is modified to for a set of NMPs $\boldsymbol{p}$ and the neural network ANN$(\cdot)$ which accepts NMPs as inputs. We define the set of priors over the NMPs as a multivariable Gaussian distribution. The mean and standard deviation on each NMP in the distribution are listed in Table III.

1. *The log-likelihood*

The log-likelihood for NS mass, radius for 2.07 and $1.4M_\odot$ neutron star are defined as follows:

$$\log\chi(M_{\mathrm{max}}) = \log\left[\frac{1}{\exp[\frac{M_{\mathrm{cal}}-M_{\mathrm{obs}}}{-0.01}]+1}\right], \tag{8}$$

$$\log\chi(R_{2.07}) = -0.5\left[\frac{R_{2.07\mathrm{cal}}-R_{2.07\mathrm{obs}}}{\Delta R_{2.07}}\right]^2 + \log[2\pi\Delta R_{2.07}^2], \tag{9}$$

$$\log\chi(R_{1.4}) = -0.5\left[\frac{R_{1.4\mathrm{cal}}-R_{1.4\mathrm{obs}}}{\Delta R_{1.4}}\right]^2 + \log[2\pi\Delta R_{1.4}^2]. \tag{10}$$

TABLE IV. The constraints imposed in the Bayesian inference: Observed maximum mass of NS, radius of $2.07M_\odot$ NS, radius of $1.4M_\odot$ NS.

Constraints		
Quantity	Value/band	Ref.
M_{max}	$>2.0M_\odot$	[91]
$R_{2.07}$	12.4 ± 1.0 km	[91]
$R_{1.4}$	13.02 ± 1.24 km	[37]

The calculations are performed for a set of NS observational properties, such as maximum mass (M_{max}), the radius at NS mass ($R_{2.07}$), and at $1.4M_\odot$ ($R_{1.4}$) listed in Table IV.

Bayesian parameter estimation is commonly carried out using the Markov chain Monte Carlo (MCMC) algorithm. This algorithm updates the existing parameters with a new set of parameters with a probability proportional to the ratio of the two points. However, MCMC approaches can lead to issues with converging to a stable posterior. To overcome this problem, we use the dynamic nested sampling algorithm [88–90]. In dynamic nested sampling, the posterior is broken into many nested "slices" with an initial n-live number of points that vary dynamically as the sampling progresses; samples are generated from each of them and then recombined to construct the posterior distribution. We use the DYNESTY dynamic nested sampling algorithm interfaced in BILBY [87] with 5000 n-live points to sample from the posterior distributions of the nuclear matter parameters. DYNESTY enables flexible Bayesian inference over complex multimodal distributions without needing to converge to the posterior before generating valid samples [90].

III. RESULTS

In this section, we present the main findings of our analysis. After selecting the best model, which we call the NS-ANN, by performing a grid search over hyperparameters, such as the model depth, model width, learning

TABLE V. RMSE on the test set, defined as $\sqrt{(1/N)\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$, where N is the a total number of samples in the test set. We also show the root-mean-square relative error $(\mathrm{RMSE}/\bar{y}) \times 100$, where $\bar{y} = (1/N)\sum_{i=1}^{N} y_i$ is the mean of the true value. All quantities are dimensionless except R_{max} and $R_{1.4}$, which are in units of kilometers, and M_{max}, which is in units of $M_\odot$.

$\hat{y}$	RMSE	$(\mathrm{RMSE}/\bar{y}) \times 100$
M_{max}	0.024	1.1
R_{max}	0.088	0.8
$R_{1.4}$	0.194	1.5
$\Lambda_{1.0}$	123.800	3.7
$\Lambda_{1.4}$	19.239	4.3
$\Lambda_{1.8}$	5.344	8.2

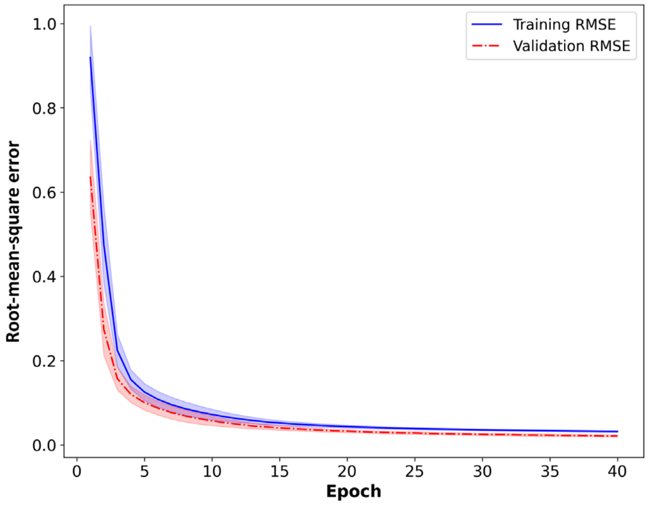

FIG. 3. Learning curves for the training (blue, solid) and validation (red, dashed) datasets plotted against the number of elapsed epochs. The shaded region around the solid curve indicates a 1σ or 68% CI region around the central value computed for ten independent training runs.

rate, etc., we determine its performance on the test set. We wish to reemphasize that the test set is never used during training or the validation phase. Evaluating the model's performance on this set quantifies the generalization capacity of the model, i.e., its predictive power on unseen data. The RMSE values obtained for each NS observable on the testing dataset are summarized in Table V. We also include the root-mean-square relative error as it provides a scale-independent measure of the generalization capacity and eases comparison between multiple dependent variables. In Fig. 3, we plot the loss as a function of training time, which is called the learning curve. The learning curve plots the root-mean-square error on the y axis against the number of elapsed epochs—or the number of full passes over the training dataset—on the x axis. We plot two different learning curves for a single training instance: one for the training loss (in blue) and the other for the validation loss (in red). Figure 3 also plots a 1σ (68% confidence interval) region centered around the training curves computed for ten independent runs to indicate the degree of variability observed by training the model on different subsets of the original dataset. The training and validation losses are the loss functions computed for the training set and validation sets, respectively. The former is a metric of how well the ANN learns the training data, while the latter indicates how well the model is able to predict by generalizing the learned information. For a NS mass of $1.4M_\odot$, we have a prediction error below 2% for the radius $R_{1.4}$ and 5% for the tidal deformability $\Lambda_{1.4}$. This implies that using the ANN, we can infer $\Lambda_{1.4}$ and $R_{1.4}$ with an average error of 19.239 and 0.194 km, respectively. The NS maximum mass $M_{\max}$ and the maximum radius $R_{\max}$ are also predicted with an error below 2%. Moreover, the saturation of the loss curves for the training and validation sets indicates that the training has converged to a minimum and that the model can generalize sufficiently well.

Once we have successfully and efficiently learned the nonlinear mapping between the empirical nuclear matter parameters and the NS observables, we must first prove the correctness of the NS-ANN model before using it in the context of a Bayesian inference framework to constrain the NMPs. In Fig. 4 we plot the 1σ confidence ellipses for Λ_M versus $L_{\mathrm{sym},0}$ and $K_{\mathrm{sym},0}$ for NS mass $M = 1.0$, 1.4, and $1.8M_\odot$ predicted by the NS-ANN for three different pseudosets of NMPs. These pseudo-NMP distributions were constructed as a multivariate Gaussian distribution

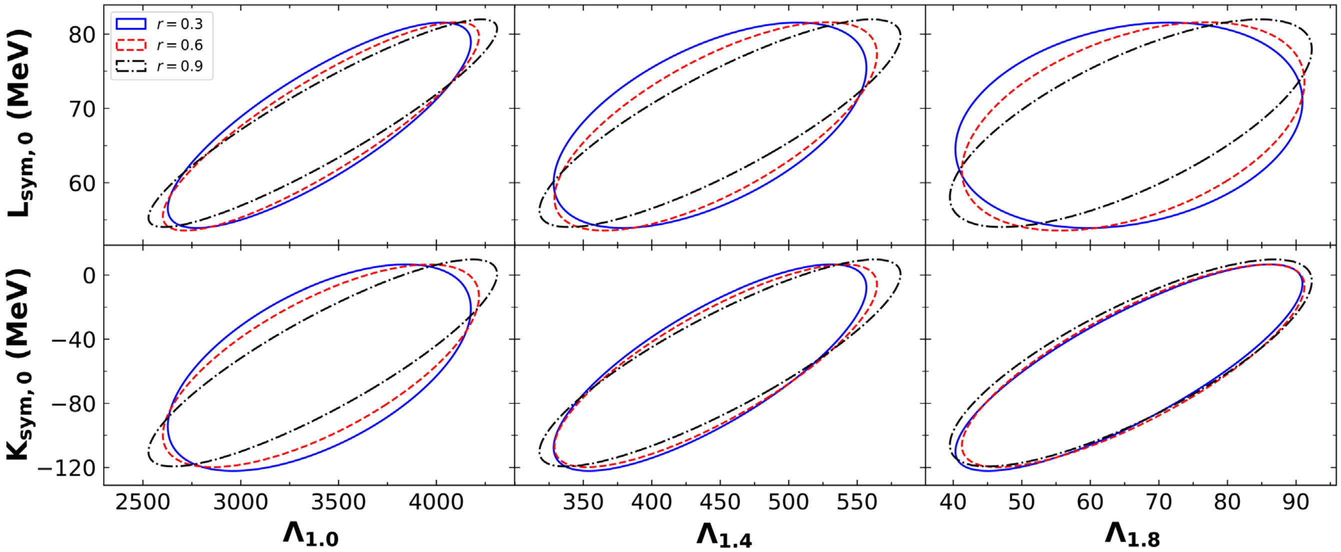

FIG. 4. The 1σ confidence ellipses in the planes of Λ_M-$L_{\mathrm{sym},0}$ (top) and Λ_M-$K_{\mathrm{sym},0}$ (bottom) with $M = 1.0$, 1.4, and $1.8M_\odot$ generated by the NS-ANN model for the correlation coefficient r between $L_{\mathrm{sym},0}$ and $K_{\mathrm{sym},0}$ set to 0.3, 0.6, and 0.9.

TABLE VI. The values of the correlation coefficients for $\Lambda_{1.0,1.4,1.8}$ with $L_{\mathrm{sym},0}$ and $K_{\mathrm{sym},0}$ for three different $L_{\mathrm{sym},0} - K_{\mathrm{sym},0}$ correlation coefficients r.

r		$\Lambda_{1.0}$	$\Lambda_{1.4}$	$\Lambda_{1.8}$
0.3	$L_{\mathrm{sym},0}$	0.81	0.56	0.23
	$K_{\mathrm{sym},0}$	0.57	0.78	0.81
0.6	$L_{\mathrm{sym},0}$	0.86	0.69	0.43
	$K_{\mathrm{sym},0}$	0.69	0.80	0.79
0.9	$L_{\mathrm{sym},0}$	0.91	0.84	0.71
	$K_{\mathrm{sym},0}$	0.86	0.86	0.79

by setting the intercorrelation between $L_{\mathrm{sym},0}$ and $K_{\mathrm{sym},0}$ to $r = 0.3$, 0.6, and 0.9, and all other parameters of the distribution to values listed in Table I. Once again, all NMPs that generate equations of state that do not satisfy the conditions from Sec. II C are discarded. We use the NS-ANN model for each NMP in the dataset to generate a corresponding set of NS observables. Through this analysis, we wish to establish that a neural network is capable of replicating the underlying microphysical information conveyed by a conventional EOS model. The values of the correlation coefficients for the results illustrated in the figure are summarized in Table VI. For the first set, we observe that the correlations of $\Lambda_{1.0,1.4,1.8}$ with $K_{\mathrm{sym},0}$ are $\chi \sim 0.5$–0.8, and those with $L_{\mathrm{sym},0}$ are $\chi \sim 0.2$–0.8. For the second set, we see a nontrivial narrowing of the confidence ellipses, indicating stronger correlations of $\Lambda_{1.0}$ and $\Lambda_{1.4}$ with $L_{\mathrm{sym},0}$ and $K_{\mathrm{sym},0}$, $\chi \sim 0.7$–0.8, while these correlations become moderate for $\Lambda_{1.8}$. We also observe that the $\Lambda_M - L_{\mathrm{sym},0}$ correlations decrease with increasing mass of NS, while an opposite trend is observed for $\Lambda_M - K_{\mathrm{sym},0}$ correlations. Moreover, for the same NS mass, $\Lambda_M - L_{\mathrm{sym},0}$ correlations are much more sensitive to $L_{\mathrm{sym},0} - K_{\mathrm{sym},0}$ correlations over $\Lambda_M - K_{\mathrm{sym},0}$ correlations. These results help emphasize that the correlations of the tidal deformability with $L_{\mathrm{sym},0}$ and $K_{\mathrm{sym},0}$ are sensitive to the physical correlations among the empirical NMPs arising from a set of physical constraints [26,27]. Additionally, the observations made in this analysis are in excellent agreement with results from previous studies performed on observing the effect of correlations of the tidal deformability with the slope of the symmetry energy and its curvature [40,43,92].

TABLE VII. The median and the minimum (min) and maximum (max) values of the associated 90% confidence intervals for the NMPs and the NS properties obtained from Skyrme, DDB, and DDB-FL datasets. We also indicate these quantities for maximum relative residual errors σ_{rr} on the NS properties.

Input					Output							
			90% CI			Skyrme			ML			
NMP		Median	Min	Max			90% CI			90% CI		
Skyrme	e_0	−16.01	−16.39	−15.59	NSP	Median	Min	Max	Median	Min	Max	σ_{rr}
	ρ_0	0.160	0.152	0.168	$M_{\max}$	2.09	1.87	2.24	2.08	1.89	2.24	2.2
	K_0	231	200	261	$R_{\max}$	10.74	9.77	11.85	10.75	9.80	11.84	1.5
	Q_0	−292	−457	−140	$R_{1.4}$	12.82	11.56	14.10	12.81	11.60	14.02	2.1
	$J_{\mathrm{sym},0}$	32	27	37	$\Lambda_{1.0}$	3279	1994	5323	3279	1973	5280	6.8
	$L_{\mathrm{sym},0}$	67	42	94	$\Lambda_{1.4}$	435	234	729	430	232	722	7.0
	$K_{\mathrm{sym},0}$	−63	−169	73	$\Lambda_{1.8}$	63	21	118	63	22	116	15.5
DDB	e_0	−16.10	−16.41	−15.80			DDB			ML		σ_{rr}
	ρ_0	0.153	0.147	0.158	$M_{\max}$	2.14	2.02	2.35	2.21	1.95	2.39	9.4
	K_0	232	201	275	$R_{\max}$	11.33	10.97	11.81	10.54	9.83	11.27	11.3
	Q_0	−108	−256	129	$R_{1.4}$	12.63	12.07	13.22	12.26	11.35	13.46	7.7
	$J_{\mathrm{sym},0}$	32	29	35	$\Lambda_{1.0}$	2802	2171	3702	2829	1869	3911	16.5
	$L_{\mathrm{sym},0}$	41	24	65	$\Lambda_{1.4}$	455	339	626	385	247	582	34.7
	$K_{\mathrm{sym},0}$	−116	−150	−73	$\Lambda_{1.8}$	79	52	125	70	28	110	65.4
DDB-FL	e_0	−16.12	−16.43	−15.83			DDB-FL			ML		σ_{rr}
	ρ_0	0.155	0.152	0.159	$M_{\max}$	2.07	2.01	2.16	2.05	1.93	2.18	6.6
	K_0	216	202	225	$R_{\max}$	11.13	10.94	11.30	10.32	9.90	10.97	10.5
	Q_0	−188	−260	−144	$R_{1.4}$	12.43	12.12	12.82	12.37	11.81	13.40	4.7
	$J_{\mathrm{sym},0}$	32	29	35	$\Lambda_{1.0}$	2612	2271	3379	2678	2146	3880	12.6
	$L_{\mathrm{sym},0}$	51	43	72	$\Lambda_{1.4}$	396	337	486	343	257	497	28.7
	$K_{\mathrm{sym},0}$	−123	−152	−70	$\Lambda_{1.8}$	62	49	76	47	27	75	64.1

At this point we also wish to note that there exists a strong model-independent intercorrelation between $L_{\mathrm{sym},0}$ and $J_{\mathrm{sym},0}$, but this correlation does not disrupt the relationships between some NS properties, such as tidal deformability, radius, and NMPs [43], and as a result it is not covered in this study.

As a next step, we apply our trained NS-ANN model to a dataset acquired from a different class of physics models in order to assess the model-dependent uncertainty related to neutron star properties.

The nuclear matter parameters are known to be model dependent and hence also the properties of stars or equations of state [93–95]. In Table VII, we compare the NS properties obtained using a dataset generated by a density-dependent meson coupling framework called the Density dependent Bayesian (DDB) model. This model is constructed using a Bayesian inference approach in conjunction with minimal constraints on the nuclear saturation properties, the maximum mass of a neutron star exceeding $2M_{\odot}$, and a low-density EOS calculated using chiral effective field theory. In the first horizontal panel, we show that NS properties predicted by the NS-ANN ML model match well with the theoretical predictions of the Skyrme model. In the second horizontal panel, then we apply our trained NS-ANN model to a dataset generated by DDB. It can be seen that the residual errors on the NS properties (NSPs) are much larger in this case. We posit that this wider uncertainty might arise due to input parameters (NMPs) in the DDB dataset lying outside the parameter space covered by the training set. To avoid any extrapolation error, we then filter the DDB according to the parameter space of the training sets. We call this set DDB-FL and apply the NS-ANN on this set.

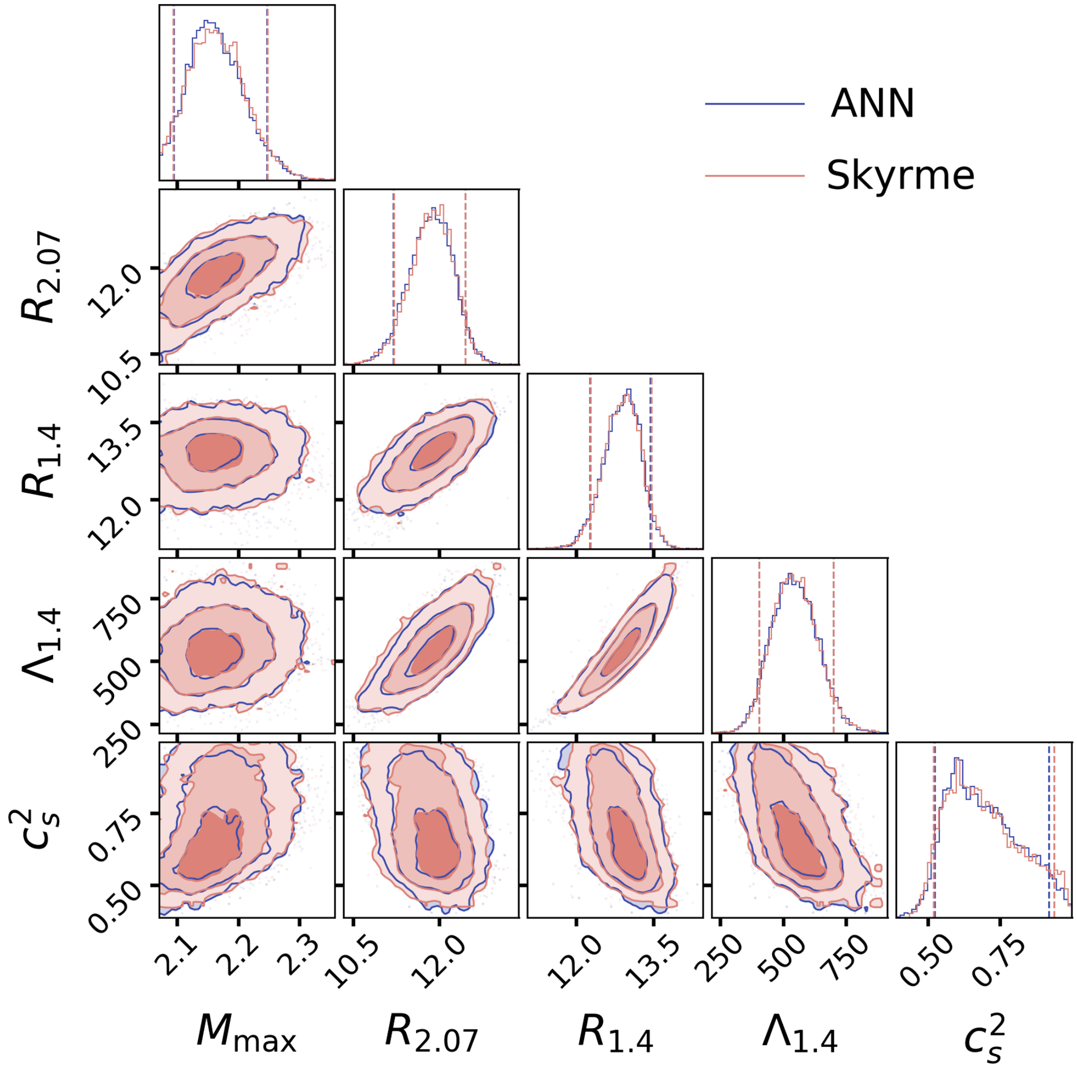

FIG. 5. Comparison of neutron star properties obtained from Bayesian posterior for prior set P1 with ANN and Skyrme model, including NS maximum mass (M_{max}), radius for a $2.07M_{\odot}$ NS, radius ($R_{1.4}$) and tidal deformability ($\Lambda_{1.4}$) for a $1.4M_{\odot}$ NS, and square of the speed of sound (C_s^2) at the center of the maximum mass NS. The vertical lines represent the 68% CIs, and the light and dark intensities represent the 1σ, 2σ, and 3σ CIs, respectively.

It can be seen that the dispersion is still large. The NS-ANN model is trained on data obtained from the nonrelativistic Skyrme model, so it is unsurprising that it fails to completely capture the features of the relativistic DDB model.

A. Validity of NS-ANN model in Bayesian inference

In this section, we apply our NS-ANN model in a Bayesian inference framework. The primary objective is to validate the trained model's effectiveness in computationally expensive calculations of this type. The results will be analyzed in two parts: (i) the posterior of neutron star properties will be compared between trained NS-ANNs and Skyrme models in Bayesian inference, and (ii) the CPU time will be compared between the two models.

B. Comparison between NS-ANN and Skyrme

As discussed above, Bayesian inference requires three ingredients: (a) the model, (b) the fit data or likelihood, and (c) prior. We will perform identical Bayesian inference calculations with both Skyrme and NS-ANN models and compare their posteriors. The input priors are the nuclear matter parameters, namely e_0, ρ_0, K_0, Q_0, $J_{\mathrm{sym},0}$, $L_{\mathrm{sym},0}$, and $K_{\mathrm{sym},0}$. In Table III, we have defined two multivariate Gaussian priors, P1 and P2. P1 represents the prior validity in which the neural network is originally trained, while in P2, we have doubled the uncertainty to evaluate the performance of the NS-ANN model. The likelihood for our calculation is defined in Sec. II D 1 for the fit data described in Sec. IV. In the likelihood, we have only considered observed NS properties because we are motivated to compare the performance of the NS-ANN trained model with the Skyrme nuclear physics model within observational constraints.

We used the PYMULTINEST sampler to sample our parameter. In Fig. 5, we compared the Bayesian posterior results for the neutron star properties obtained from the prior set P1 with the NS-ANN and Skyrme model.

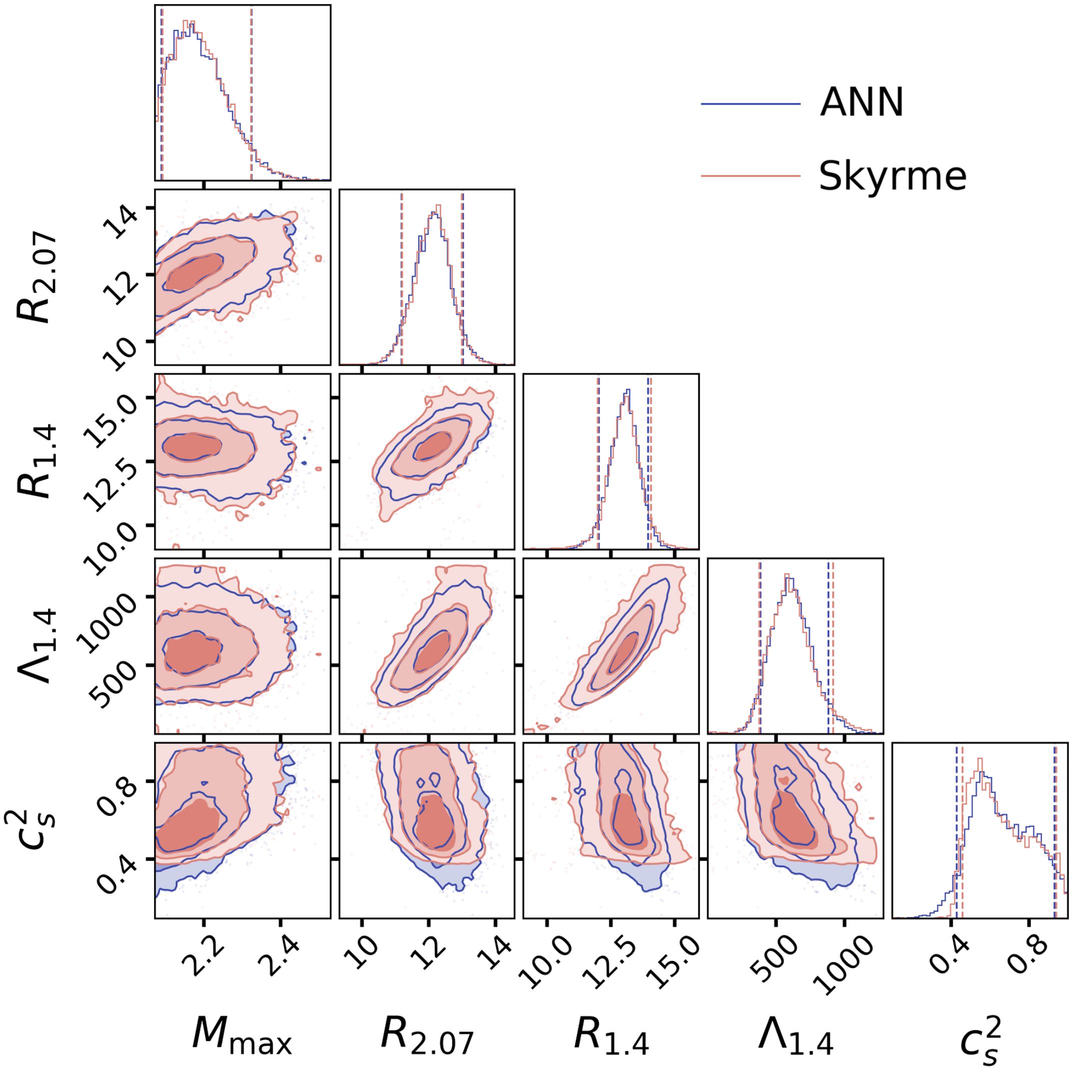

FIG. 6. Same as Fig. 5 but for prior set P2.

This included the NS maximum mass ($M_{\max}$), radius for a $2.07M_{\odot}$ NS, radius ($R_{1.4}$), tidal deformability ($\Lambda_{1.4}$) for a $1.4M_{\odot}$ NS, and the square of the speed of sound (C_s^2) at the center of the maximum mass NS. We also compared the same for the prior set P2 in Fig. 6. In both figures, the joint distribution of NS properties is represented by corner plots, which are obtained from Bayesian inference posteriors. Each variable in a corner plot is plotted on one of the axes, and its distribution is shown by its diagonal. Off-diagonal plots show the joint distributions of each pair of variables. Using it to understand how variables are related is a useful tool. Now, it is time to compare their accuracy and validity.

In Table VIII, we compared the relative % residuals of the NS-ANN model with the Skyrme model for both P1 and P2. It is worth noting that the results for P1 are very comparable to the original physics model. This is expected because P1 is the prior range in which the NS-ANN was originally trained. The relative % residuals for the NS maximum mass, radius for 2.07 and $1.4M_{\odot}$ NS, and tidal deformability for $1.4M_{\odot}$ NS are ∼1%, while the square of the speed of sound is ∼3% at the center of the maximum mass NS. However, in the case of P2, all the residuals are noticeably increased, with the tidal deformability of $1.4M_{\odot}$ NS and the square of the speed of sound at the center of the maximum mass NS being around ∼5%. It should be emphasized that the neural network model can be trusted for interpolation but used with caution for extrapolation. The main advantage of this type of ANN model is that it can produce very fast draft results for computationally expensive calculations.

C. CPU Time

In Table IX, we compared the CPU time for NS-ANN and Skyrme models for both prior sets. The timing tests were performed on a 12-core Intel i7-8700K CPU @ 3.70 GHz. It is noteworthy that the NS-ANN model is approximately 500 times faster than the Skyrme model. It is exciting that the neural network model can be employed in computationally expensive calculations to obtain fast draft results. This will help in reducing carbon emissions and help reduce the environmental cost of using high computational power for the preliminary testing of any model.

TABLE VIII. The maximum relative residual for the maximum mass of a NS, denoted as Mmax, is attained. Additionally, we determine the radius R2.07 for a 2.07 solar mass ($M_{\odot}$) NS, the radius R1.4, and the dimensionless tidal deformability $\Lambda_{1.4}$ for a $1.4M_{\odot}$ NS, as well as the squared speed of sound at the core of the NS with maximum mass. These quantities are obtained using the prior sets P1 and P2.

		$\mathrm{M}_{\max}$	$\mathrm{R}_{2.07}$	$\mathrm{R}_{1.4}$	$\Lambda_{1.4}$	C_s^2
NS Units		$M_{\odot}$	km	km	⋯	c^2
Maximum relative % residual	P1	0.3	0.6	0.3	1.2	2.5
	P2	0.8	1.6	1.5	5.3	5.1

TABLE IX. CPU inference time estimates for the ANN model and a Skyrme model to infer NS observations from a set of NMPs for prior sets P1 and P2. The timing tests were performed on a 12-core Intex i7-8700K CPU @ 3.70 GHz. The inference is performed with a batch size of one.

Model	P1	P2
ANN	2.23 min	3.08 min
Skyrme	16 h 27 min	19 h 55 min

IV. CONCLUSIONS

We have demonstrated the application of ANNs to analyze NS observables like the radius, mass, and tidal deformability from a set of seven parameters, or nuclear saturation properties that characterize the equation of state of cold dense matter. We have shown that a neural network that models such a mapping is able to learn the microphysics information of finite nuclei, which are the intercorrelations arising between the NMPs. Moreover, we delved into the utilization of ANNs as a complement to theoretical models, enabling them to undertake computationally demanding tasks, such as Bayesian inference studies to constrain nuclear models. This study demonstrates that neural networks or some advanced AI tools not only function as efficient surrogates for traditional physics-based EOS models but also have the potential to eventually replace phenomenological models, particularly for computationally intensive or high-dimensional problems.

Let us briefly summarize what we have done in this work. First, we generate a set of pseudo-NMP data using the Skyrme model. A suitable set of NMPs are chosen so that the resulting neutron star equations of state are consistent with the currently observed maximum mass of $\sim 2M_{\odot}$ and satisfy causality constraints. Next, we train an artificial neural network on this dataset to learn a nonlinear mapping between the set of nuclear matter parameters and NS observables. We demonstrate that the ANN is capable of inferring, with reasonable accuracy, NS observables from empirical parameters, as compared to conventional physics models which require the computation of an EOS followed by tedious equation solving. This NS-ANN model is then validated to ensure that it satisfies the microphysics information of finite nuclear matter. More specifically, we study whether a trained ANN model is able to capture correlations between the tidal deformability for a NS with mass $1.4M_{\odot}$, $\Lambda_{1.4}$ and the symmetry energy slope $L_{\mathrm{sym},0}$ as well as its curvature $K_{\mathrm{sym},0}$. We find that the NS-ANN model learns a mapping that is sensitive to $L_{\mathrm{sym},0} - K_{\mathrm{sym},0}$ correlations in agreement with previous studies. Using the Bayesian inference framework, we determine the extent to which the neural network can replicate the physics model and then utilize the trained neural network for inference tasks, which typically involve computationally expensive calculations. By doing so, we are able to achieve results that are comparable to the physics model while significantly reducing the computational time required.

Presently, our framework is a proof of concept that demonstrates the applicability of ANNs to NS physics. For a more realistic application of our framework, empirical uncertainties ought to be considered. This can be achieved, for example, by using a class of ANNs called Bayesian neural networks to perform the inference task. Bayesian networks have the ability to cast the problem into a probabilistic domain by inferring probability distributions over a prior of NMP inputs. A model that predicts uncertainties will also be able to potentially further reveal the effects of NS observational constraints on the NMPs. Presently, the entire domain of validity of the ANN cannot be automatically ensured. Due to a lack of training data for some parts in the modeled output space, the ANN might be unable to accurately predict regions that have not been learned.

While the current study employs traditional methods and neural network models for Bayesian analysis, we recognize the emerging potential of more sophisticated techniques. Specifically, normalizing flows and convolutional variational autoencoders (CVAEs) offer promising avenues for future work. Normalizing flows and CVAEs are powerful methods able to model complex probability distributions [96]. Although normalizing flows provide an accurate evaluation of likelihood, they can be computationally demanding, especially when dealing with high-dimensional data. On the other hand, CVAEs are efficient and can handle large datasets, but only offer approximate likelihood evaluations. They may be useful when handling gridlike data, such as images. The decision to use either method depends on task requirements, such as exact likelihood estimation and computational efficiency.

This work is limited only to hadronic NS compositions, i.e., the set of β-equilibrated equations of state employed in this work is composed of neutrons, protons, electrons, and muons. Present observational constraints on NS properties cannot rule out the possibility of exotic degrees of freedom or deconfined quark phases inside the NS core.

In future work, a detailed and systematic analysis with an ANN trained on a set of equations of state with different compositions of particles is required to investigate the uncertainties on higher-order NMPs with available observational constraints. The exploration of more advanced AI methods, such as normalizing flows and CVAEs, not only holds the promise for enhancing computational efficiency, but also raises the possibility of becoming the *de facto* method for certain types of nuclear and astrophysical analyses.

The trained network and corresponding datasets are made publicly available through the GitHub repository [97].

ACKNOWLEDGMENTS

T. M. would like to thank national funds from FCT (Fundação para a Ciência e a Tecnologia, I.P, Portugal) under Projects No. UID-FIS-04564-2019, No. UIDP-04564-2020, No. UIDB-04564-2020, and No. POCI-01-0145-FEDER-029912 with financial support from Science, Technology, and Innovation, in its FEDER component, and by the FCT/MCTES budget through national funds (OE). The authors acknowledge the Laboratory for Advanced Computing at the University of Coimbra for providing high-performance computer resources that have contributed to the research results reported within this paper [98].

[1] N. K. Glendenning, *Compact Stars: Nuclear Physics, Particle Physics and General Relativity* (Springer Science & Business Media, New York, 2012).

[2] P. Haensel, A. Y. Potekhin, and D. G. Yakovlev, *Neutron Stars 1: Equation of State and Structure* (Springer, New York, 2007), Vol. 326.

[3] E. R. Most, L. R. Weih, L. Rezzolla, and J. Schaffner-Bielich, Phys. Rev. Lett. **120**, 261103 (2018).

[4] S. B. Ruester, M. Hempel, and J. Schaffner-Bielich, Phys. Rev. C **73**, 035804 (2006).

[5] R. C. Tolman, Phys. Rev. **55**, 364 (1939).

[6] J. R. Oppenheimer and G. M. Volkoff, Phys. Rev. **55**, 374 (1939).

[7] K. Hebeler and A. Schwenk, Phys. Rev. C **82**, 014314 (2010).

[8] I. Tews, T. Krüger, K. Hebeler, and A. Schwenk, Phys. Rev. Lett. **110**, 032504 (2013).

[9] G. Hagen, T. Papenbrock, A. Ekström, K. A. Wendt, G. Baardsen, S. Gandolfi, M. Hjorth-Jensen, and C. J. Horowitz, Phys. Rev. C **89**, 014319 (2014).

[10] A. Roggero, A. Mukherjee, and F. Pederiva, Phys. Rev. Lett. **112**, 221103 (2014).

[11] P. G. Krastev, Galaxies **10**, 16 (2022).

[12] B. A. Freedman and L. D. McLerran, Phys. Rev. D **16**, 1130 (1977).

[13] A. Kurkela, P. Romatschke, and A. Vuorinen, Phys. Rev. D **81**, 105021 (2010).

[14] T. Gorda, A. Kurkela, P. Romatschke, M. Säppi, and A. Vuorinen, Phys. Rev. Lett. **121**, 202701 (2018).

[15] Y. Fujimoto, K. Fukushima, and K. Murase, J. High Energy Phys. 03 (2021) 273.

[16] G. Aarts, J. Phys. Conf. Ser. **706**, 022004 (2016).

[17] G. A. Lalazissis, J. Konig, and P. Ring, Phys. Rev. C **55**, 540 (1997).

[18] B. G. Todd-Rutel and J. Piekarewicz, Phys. Rev. Lett. **95**, 122501 (2005).

[19] P. Klupfel, P. G. Reinhard, T. J. Burvenich, and J. A. Maruhn, Phys. Rev. C **79**, 034310 (2009).

[20] A. Sulaksono, T. J. Buervenich, P. G. Reinhard, and J. A. Maruhn, Phys. Rev. C **79**, 044306 (2009).

[21] N.-B. Zhang, B.-A. Li, and J. Xu, Astrophys. J. **859**, 90 (2018).

[22] B.-J. Cai and B.-A. Li, Phys. Rev. C **103**, 034607 (2021).

[23] H. Gil, Y.-M. Kim, P. Papakonstantinou, and C. H. Hyun, Phys. Rev. C **103**, 034330 (2021).

[24] E. Annala, T. Gorda, E. Katerini, A. Kurkela, J. Nättilä, V. Paschalidis, and A. Vuorinen, Phys. Rev. X **12**, 011058 (2022).

[25] S. Altiparmak, C. Ecker, and L. Rezzolla, Astrophys. J. Lett. **939**, L34 (2022).

[26] J. Margueron, R. Hoffmann Casali, and F. Gulminelli, Phys. Rev. C **97**, 025805 (2018).

[27] J. Margueron, R. Hoffmann Casali, and F. Gulminelli, Phys. Rev. C **97**, 025806 (2018).

[28] P. Demorest, T. Pennucci, S. Ransom, M. Roberts, and J. Hessels, Nature (London) **467**, 1081 (2010).

[29] E. Fonseca *et al.*, Astrophys. J. **832**, 167 (2016).

[30] Z. Arzoumanian *et al.* (NANOGrav Collaboration), Astrophys. J. Suppl. Ser. **235**, 37 (2018).

[31] J. Antoniadis *et al.*, Science **340**, 6131 (2013).

[32] E. Fonseca *et al.*, Astrophys. J. Lett. **915**, L12 (2021).

[33] R. W. Romani, D. Kandel, A. V. Filippenko, T. G. Brink, and W. Zheng, Astrophys. J. Lett. **908**, L46 (2021).

[34] J. Faber, Classical Quantum Gravity **26**, 114004 (2009).

[35] M. D. Duez, Classical Quantum Gravity **27**, 114002 (2010).

[36] T. E. Riley *et al.*, Astrophys. J. Lett. **887**, L21 (2019).

[37] M. C. Miller *et al.*, Astrophys. J. Lett. **887**, L24 (2019).

[38] T. E. Riley *et al.*, Astrophys. J. Lett. **918**, L27 (2021).

[39] T. Malik, M. Ferreira, B. K. Agrawal, and C. Providência, Astrophys. J. **930**, 17 (2022).

[40] F. J. Fattoyev, J. Piekarewicz, and C. J. Horowitz, Phys. Rev. Lett. **120**, 172702 (2018).

[41] Z. Carson, A. W. Steiner, and K. Yagi, Phys. Rev. D **99**, 043010 (2019).

[42] S. Ghosh, B. K. Pradhan, D. Chatterjee, and J. Schaffner-Bielich, Front. Astron. Space Sci. **9**, 864294 (2022).

[43] T. Malik, B. K. Agrawal, C. Providência, and J. N. De, Phys. Rev. C **102**, 052801 (2020).

[44] E. Annala, T. Gorda, A. Kurkela, J. Nättilä, and A. Vuorinen, Nat. Phys. **16**, 907 (2020).

[45] K. Hebeler, J. M. Lattimer, C. J. Pethick, and A. Schwenk, Astrophys. J. **773**, 11 (2013).

[46] C. Drischler, K. Hebeler, and A. Schwenk, Phys. Rev. C **93**, 054314 (2016).

[47] J. S. Read, B. D. Lackey, B. J. Owen, and J. L. Friedman, Phys. Rev. D **79**, 124032 (2009).

[48] F. Ozel, D. Psaltis, T. Guver, G. Baym, C. Heinke, and S. Guillot, Astrophys. J. **820**, 28 (2016).

[49] C. A. Raithel, F. Özel, and D. Psaltis, Astrophys. J. **844**, 156 (2017).

[50] L. Lindblom, Phys. Rev. D **82**, 103011 (2010).

[51] L. Lindblom and N. M. Indik, Phys. Rev. D **86**, 084003 (2012).

[52] R. Essick, P. Landry, and D. E. Holz, Phys. Rev. D **101**, 063007 (2020).

[53] P. Landry, R. Essick, and K. Chatziioannou, Phys. Rev. D **101**, 123007 (2020).

[54] M.-Z. Han, J.-L. Jiang, S.-P. Tang, and Y.-Z. Fan, Astrophys. J. **919**, 11 (2021).

[55] K. He, X. Zhang, S. Ren, and J. Sun, in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Las Vegas, USA, 2016), pp. 770–778, 10.1109/CVPR.2016.90.

[56] T. Young, D. Hazarika, S. Poria, and E. Cambria, IEEE Comput. Intell. Mag. **13**, 55 (2018).

[57] D. George and E. A. Huerta, Phys. Rev. D **97**, 044039 (2018).

[58] D. George and E. A. Huerta, Phys. Lett. B **778**, 64 (2018).

[59] M. Carrillo, M. Gracia-Linares, J. A. González, and F. S. Guzmán, Gen. Relativ. Gravit. **48**, 141 (2016).

[60] A. J. Larkoski, I. Moult, and B. Nachman, Phys. Rep. **841**, 1 (2020).

[61] D. Guest, K. Cranmer, and D. Whiteson, Annu. Rev. Nucl. Part. Sci. **68**, 161 (2018).

[62] D. Bourilkov, Int. J. Mod. Phys. A **34**, 1930019 (2020).

[63] P. Bedaque, A. Boehnlein, M. Cromaz, M. Diefenthaler, L. Elouadrhiri, T. Horn, M. Kuchera, D. Lawrence, D. Lee, S. Lidia *et al.*, arXiv:2006.05422.

[64] Y. Fujimoto, K. Fukushima, and K. Murase, Phys. Rev. D **101**, 054016 (2020).

[65] F. Morawski and M. Bejger, Astron. Astrophys. **642**, A78 (2020).

[66] M. U. Anil, K. Banerjee, T. Malik, and C. Providência, J. Cosmol. Astropart. Phys. 01 (2022) 045.

[67] Y. Fujimoto, K. Fukushima, and K. Murase, Phys. Rev. D **98**, 023019 (2018).

[68] I. Vidana, C. Providencia, A. Polls, and A. Rios, Phys. Rev. C **80**, 045806 (2009).

[69] M. a. Ferreira and C. Providência, J. Cosmol. Astropart. Phys. 07 (2021) 011.

[70] T. Hinderer, Astrophys. J. **677**, 1216 (2008).

[71] K. Hornik, M. Stinchcombe, and H. White, Neural Netw. **2**, 359 (1989).

[72] G. Cybenko, Math. Control. Signals, Syst. **2**, 303 (1989).

[73] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, in *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision* (IEEE, Kyoto, Japan, 2009), pp. 2146–2153, 10.1109/ICCV.2009.5459469.

[74] V. Nair and G. E. Hinton, in *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10* (Omnipress, Madison, WI, 2010), pp. 807–814, ISBN: 9781605589077.

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

[76] W. Wei, E. A. Huerta, M. Yun, N. Loutrel, M. A. Shaikh, P. Kumar, R. Haas, and V. Kindratenko, Astrophys. J. **919**, 82 (2021).

[77] A. J. K. Chua and M. Vallisneri, Phys. Rev. Lett. **124**, 041102 (2020).

[78] T. Chen and C. Guestrin, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16* (ACM, New York, NY, 2016), pp. 785–794, ISBN 978-1-4503-4232-2, 10.1145/2939672.2939785.

[79] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, J. Mach. Learn. Res. **12**, 2825 (2011).

[80] Y. Lecun, Y. Bengio, and G. Hinton, Nat. Cell Biol. **521**, 436 (2015).

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

[82] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, OSDI'16* (USENIX Association, Berkeley, CA, USA, 2016), p. 265–283, ISBN: 9781931971331.

[83] X. Glorot and Y. Bengio, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Vol. 9 of Proceedings of Machine Learning Research (PMLR, 2010), pp. 249–256, https://proceedings.mlr.press/v9/glorot10a.html.

[84] D. P. Kingma and J. Ba, in *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 2015, Conference Track Proceedings*, edited by Y. Bengio and Y. LeCun (ICLR, CA, USA, 2015), arXiv:1412.6980.

[85] S. Wesolowski, N. Klco, R. J. Furnstahl, D. R. Phillips, and A. Thapaliya, J. Phys. G **43**, 074001 (2016).

[86] R. J. Furnstahl, N. Klco, D. R. Phillips, and S. Wesolowski, Phys. Rev. C **92**, 024005 (2015).

[87] G. Ashton *et al.*, Astrophys. J. Suppl. Ser. **241**, 27 (2019).

[88] J. Skilling, in *Bayesian Inference and Maximum Entropy Methods in Science and Engineering: 24th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, edited by R. Fischer, R. Preuss, and U. V. Toussaint, Vol. 735 of American Institute of Physics Conference Series (American Institute of Physics, Washington, DC, USA, 2004), pp. 395–405, 10.1063/1.1835238.

[89] E. Higson, W. Handley, M. Hobson, and A. Lasenby, Stat. Comput. **29**, 891 (2019).

[90] J. S. Speagle, Mon. Not. R. Astron. Soc. **493**, 3132 (2020).

[91] M. C. Miller *et al.*, Astrophys. J. Lett. **918**, L28 (2021).

[92] M. Ferreira, M. Fortin, T. Malik, B. K. Agrawal, and C. Providência, Phys. Rev. D **101**, 043021 (2020).

[93] B. K. Pradhan, D. Chatterjee, R. Gandhi, and J. Schaffner-Bielich, Nucl. Phys. **A1030**, 122578 (2023).

[94] T. Malik, M. Ferreira, M. B. Albino, and C. Providência, Phys. Rev. D **107**, 103018 (2023).

[95] J. Zhou, J. Xu, and P. Papakonstantinou, Phys. Rev. C **107**, 055803 (2023).

[96] R. Morrow and W. Chiu, https://arxiv.org/abs/2004.05617.

[97] A. Thete, GitHub (2022), https://github.com/ameya1101/NS-ANN.

[98] https://www.uc.pt/lca.