

From neutron star observations to nuclear matter properties: A machine learning approach

Valéria Carvalho^{✉,*}, Márcio Ferreira^{✉,†} and Constança Providência^{✉,‡}
CFisUC, Department of Physics, University of Coimbra, P-3004-516 Coimbra, Portugal

 (Received 7 January 2024; accepted 30 May 2024; published 26 June 2024)

This study is devoted to the inference problem of extracting the nuclear matter properties directly from a set of mass-radius observations. We employ Bayesian neural networks (BNNs), which is a probabilistic model capable of estimating the uncertainties associated with its predictions. To simulate different noise levels on the $M(R)$ observations, we create three different sets of mock data. Our results show BNNs as an accurate and reliable tool for predicting the nuclear matter properties whenever the true values are not completely outside the training dataset statistics, i.e., if the model is not heavily dependent on its extrapolating capacities. Using real mass-radius pulsar data, the model predicted, for instance, $L_{\text{sym}} = 39.80 \pm 17.52$ MeV and $K_{\text{sym}} = -101.67 \pm 62.86$ MeV (2σ interval). Our study provides a valuable inference framework when new neutron star data becomes available.

DOI: [10.1103/PhysRevD.109.123038](https://doi.org/10.1103/PhysRevD.109.123038)

I. INTRODUCTION

The properties and composition of matter properties inside neutron stars (NSs) remain an open question. The equation of state (EOS) of dense and asymmetric nuclear matter, realized inside NSs, is the key quantity of research in NS physics. Constraints on the EOS at moderate and high baryonic densities are mainly supplied by observations of massive NSs (M/M_{\odot}): 1.908 ± 0.016 for PSR J1614 – 2230 [1–3], 2.01 ± 0.04 for PSR J0348–0432 [4], 2.08 ± 0.07 for PSR J0740 + 6620 [5], and 2.13 ± 0.04 for J1810 + 1714 [6]. Furthermore, the recent advent of multimessenger astrophysics combined different sources of information regarding NS physics (e.g., gravitational waves, photons, and neutrinos). The observation of compact binary coalescence events such as GW170817 [7] and GW190425 [8] by LIGO/Virgo collaboration posed further constraints on the EOS. The recent inferences of the PSR J0030 + 045 pulsar mass [9,10] and PSR J0740 + 6620 radius [11–13] by NICER (Neutron star Interior Composition Explorer) experiment also narrowed down the possible NS matter scenarios. Observations from future experiments, with higher precision measurements, such as the enhanced x-ray Timing and Polarimetry mission (eXTP) [14,15], the (STROBE-X) [16], and Square Kilometer Array [17] telescope, will be fundamental in restricting the different scenarios for the NS matter properties.

The low density region of the EOS is constrained by chiral effective field theory (cEFT) [18,19], and at high

densities, pQCD is reliable (for a review [20]). The inference of the EOS of NSs given a set of theoretical and/or observational constraints is normally implemented via Bayesian inference scheme, see, e.g., [21–23]. Another interesting method also implemented is the Gaussian processes [24–27]. Let us point out that several approaches have considered agnostic descriptions of the EOS subjected to the above constraints, in particular, in several works the two extreme constraining EOS are connected using a piecewise polytropic, speed of sound or spectral interpolation, and if necessary, causality is imposed [28–33]. These descriptions do not allow, however, for the determination of the nuclear matter properties.

The use of neural networks (NNs) based inference frameworks in high energy physics has gathered significant attention across various disciplines, as highlighted by the comprehensive review [34]. Our recent contribution [35] focused on employing a Bayesian neural network (BNN) to map NS observations to the speed of sound squared and proton fraction inside NSs. Additionally, we explored NNs in mapping the EOS of β -equilibrium NS matter to the properties of nuclear matter in [36], where we tested our final model with 31 nuclear models. Expanding beyond our contributions, several studies have been conducted [26,37–47]. For instance, deep NNs are explored to deduce nuclear matter properties in [43,44] using two architectures: one mapping $M(R)$ curves to the EOS and the other mapping the EOS to nuclear matter properties. However, the need for robust uncertainty quantification is evident in both articles. Addressing questions like “How confident is a model about its predictions?” remains pivotal.

The present work aims to directly map a set of NS observables (mass-radius observations) to nuclear matter

*val.mar.dinis@uc.pt

†marcio.ferreira@uc.pt

‡cp@uc.pt

properties using an inference framework based on BNNs. Training on a dataset of relativistic mean field (RMF) models constrained by minimal conditions such as low-density properties and pure-neutron matter, our trained model facilitates instantaneous inference for diverse nuclear matter properties across various observation sets. This approach eliminates the need for separate Bayesian analyses for each observation set.

The paper is organized as follows. A basic introduction to BNNs is presented in Sec. II. Then the family of nuclear models chosen and the generation of the mock observation datasets is explained in Sec. III. The model results for the properties of nuclear matter are discussed in Sec. IV, where the section is further divided into three tests, and lastly, the conclusions are drawn in Sec. V.

II. BAYESIAN NEURAL NETWORKS

Bayesian neural networks (BNNs) represent a Bayesian approach framework designed to quantify both aleatoric and epistemic uncertainties within a dataset [48]. They can be viewed as an extension of traditional NNs wherein the model weights are stochastic and follow a specified probability distribution. The prediction of BNNs, denoted as \mathbf{y}^* for a given input \mathbf{x}^* , transforms into a Bayesian model average

$$P(\mathbf{y}^*|\mathbf{x}^*, D) = \int_{\theta} P(\mathbf{y}^*|\mathbf{x}^*, \theta) P(\theta|D) d\theta, \quad (1)$$

where θ denotes the model's weights, D the dataset, $P(\mathbf{y}^*|\mathbf{x}^*, \theta)$ is the network's distribution (which captures the noise present in the data), and $P(\theta|D)$ is the posterior distribution of our weights (which characterizes the model uncertainty). Both aleatoric and epistemic uncertainties are

captured by, respectively, $P(\mathbf{y}^*|\mathbf{x}^*, \theta)$ and $P(\theta|D)$ (see Ref. [49] for details).

Variational inference method is used to solve Eq. (1), where the real posterior $P(\theta|D)$ is approximated by a variational posterior $q_{\phi}(\theta)$. The Kullback-Leibler relation, which measures the dissimilarity between two probability distributions, is used to quantify the quality of the approximation (i.e., how close is the variational posterior to the real posterior),

$$\text{KL}(q_{\phi}(\theta)||P(\theta|D)) = \int_{\theta} q_{\phi}(\theta) \log\left(\frac{q_{\phi}(\theta)}{P(\theta|D)}\right) d\theta. \quad (2)$$

The variational posterior is determined by optimizing the following objective function (see Ref. [35] for details):

$$q_{\phi}^* = \arg \min_{q_{\phi}} [\text{KL}(q_{\phi}(\theta)||P(\theta)) - \mathbb{E}_{q_{\phi}(\theta)}(\log P(D|\theta))]. \quad (3)$$

Choosing a multivariate Gaussian for the variational posterior, $q_{\phi}(\theta) = \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$, and a multivariate Gaussian with diagonal covariance matrix for the prior, $P(\theta) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, and using Monte Carlo sampling to calculate the expected values of our target objective function, we obtain

$$F(D, \phi) = \frac{1}{2D_s} [(-\log \det(\boldsymbol{\Sigma}_q)) - k + \text{tr}(\boldsymbol{\Sigma}_q) + (\boldsymbol{\mu}_q)^T (\boldsymbol{\mu}_q)] - \frac{1}{B} \sum_{i=1}^B \frac{1}{N} \sum_{n=1}^N \log P(y_i|x_i, \theta^{(n)}), \quad (4)$$

where $\theta^{(n)}$ are samples from the variational posterior, $q_{\phi}(\theta)$, B is the number of points of the minibatch, D_s is the number

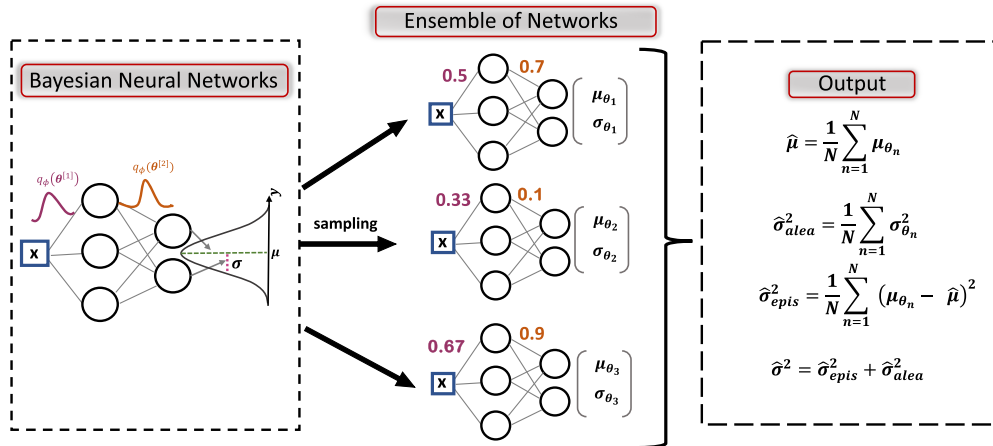


FIG. 1. Intuitive illustration highlighting the predicted uncertainty quantification obtained by the optimized BNN model. The mean vector $\hat{\boldsymbol{\mu}}$ and the variance vector $\hat{\boldsymbol{\sigma}}^2$ of the predicting distribution $P(\mathbf{y}^*|\mathbf{x}^*, D)$ are calculated, for a fixed \mathbf{x}^* , utilizing the law of total expectation and total variance. These principles are elucidated more comprehensively in our earlier work [35]. The equation governing $\hat{\boldsymbol{\mu}}$ describes the mean vector obtained through the law of total expectation. Simultaneously, the variance vector, denoted by $\hat{\boldsymbol{\sigma}}^2$ and obtained via the law of total variance, is intricately partitioned into two distinctive components; an epistemic component ($\hat{\boldsymbol{\sigma}}_{\text{epis}}^2$) and an aleatoric component ($\hat{\boldsymbol{\sigma}}_{\text{alea}}^2$).

of points of the training dataset, k is the dimension of the identity matrix of the prior and N is the number of samples we take off the variational posterior (we used $N = 10^4$). Once the network is trained and the mean μ_q and covariance matrix Σ_q are found, which best approximate the true posterior, we solve Eq. (1) via Monte Carlo sampling,

$$P(\mathbf{y}^*|\mathbf{x}^*, D) = \int_{\theta} P(\mathbf{y}^*|\mathbf{x}^*, \theta) q_{\phi}(\theta) d\theta \quad (5)$$

$$= \frac{1}{N} \sum_{n=1}^N P(\mathbf{y}^*|\mathbf{x}^*, \theta^{(n)}), \quad \theta^{(n)} \sim q_{\phi}(\theta). \quad (6)$$

The meaning of the last equation is schematically illustrated in Fig. 1. Each sampled weights $\theta^{(n)}$ corresponds to a specific NN, and the BNNs output is determined by the ensemble prediction.

III. DATASET

A. Nuclear models

The data we use for training the BNN models consists of nuclear models based on a relativistic mean field (RMF) description, where the nucleon interaction is mediated through the exchange of scalar-isoscalar mesons, vector-isoscalar mesons, and vector-isovector mesons. The model details are omitted here but can be seen in [23]. The dataset consisting of 25 287 nuclear models was obtained by imposing minimal constraints on several nuclear saturation properties, reproducing $2M_{\odot}$ NSs, and a consistent low-density pure neutron matter with N³LO calculations in chiral effective field theory (see Table I).

To a good approximation, the EOS of nuclear matter can be written in terms of the binding energy per nucleon $\epsilon(n, \delta)$ and can be decomposed into a symmetric and asymmetric part [55,56],

$$\epsilon(n, \delta) \approx \epsilon_{\text{SNM}}(n) + E_{\text{sym}}(n)\delta^2 + \dots, \quad (7)$$

TABLE I. Constraints on the binding energy per nucleon ϵ_0 , incompressibility K_0 , and symmetry energy J_{sym} at the nuclear saturation density n_0 (with 1σ uncertainties). The pressure of pure neutron matter (PNM) is considered at densities of 0.08 fm^{-3} , 0.12 fm^{-3} , and 0.16 fm^{-3} , obtained from a χ EFT calculation [18].

		Constraints		
	Quantity		Value/band	Ref
NMP	(fm^{-3})	n_0	0.153 ± 0.005	[50]
		ϵ_0	-16.1 ± 0.2	[51]
	(MeV)	K_0	230 ± 40	[52,53]
		$J_{\text{sym},0}$	32.5 ± 1.8	[54]
PNM (MeV fm^{-3})		$P(\rho)$	$2 \times \text{N}^3\text{LO}$	[18]
		$dP/d\rho$	>0	
NS mass (M_{\odot})		M_{max}	>2.0	[5]

where $\delta = (n_n - n_p)/n$ is the isospin asymmetry, n_p and n_n are the proton and neutron density, respectively, $n = n_p + n_n$ is the baryonic density, $\epsilon_{\text{SNM}}(n) = \epsilon(n, \delta = 0)$ represents the binding energy per nucleon of symmetric nuclear matter, and $E_{\text{sym}}(n)$ denotes the symmetry energy, which could be written as

$$E_{\text{sym}}(n) = \left. \frac{\partial^2 \epsilon(n, \delta)}{2\partial\delta^2} \right|_{\delta=0}. \quad (8)$$

Expanding both symmetric and asymmetric parts in a Taylor series around the saturation density n_0 until third order, we get

$$\begin{aligned} \epsilon(n, \delta) \approx & \underbrace{\epsilon_0 + \frac{K_0}{2}x^2 + \frac{Q_0}{6}x^3}_{\epsilon_{\text{SNM}}(n)} \\ & + \underbrace{\left(J_{\text{sym}} + L_{\text{sym}}x + \frac{K_{\text{sym}}}{2}x^2 + \frac{Q_{\text{sym}}}{6}x^3 \right)}_{E_{\text{sym}}(n)} \delta^2 + \dots, \end{aligned} \quad (9)$$

where $x = (n - n_0)/(3n_0)$. The $\epsilon_0 = \epsilon_{\text{SNM}}(n_0)$ term denotes the binding energy per particle, and J_{sym} the symmetry energy, K_0 is the incompressibility coefficient, Q_0 the skewness, while L_{sym} , K_{sym} , Q_{sym} are, respectively, the slope, curvature and skewness of symmetry energy at saturation density, respectively. The dataset statistics for the nuclear matter properties are in Table II. A detailed examination of these parameters, including asymmetric parameters with higher order effects in Eq. (7), can be found in [57].

B. Structure

Our objective involves utilizing simulated observations of neutron stars (referred to as the set \mathbf{X}) as inputs for a BNN model, denoted as $P(\mathbf{Y}|\mathbf{X}, \theta)$, where θ represents parameters sampled from the variational posterior distribution of the model, i.e., $\theta \sim q_{\phi}(\theta)$. The primary objective is to establish a probability distribution for the output space, denoted as the set \mathbf{Y} . This set encompasses properties of nuclear matter, introduced in the preceding section and detailed in Table II. However, our focus is specifically on training and predicting the incompressibility K_0 and skewness Q_0 of symmetric nuclear matter, and the symmetry energy J_{sym} , along with its slope L_{sym} , curvature K_{sym} , and skewness Q_{sym} . We are not predicting the values of n_0 and E_0 because they are theoretical and experimental well restricted and have been strongly constrained when the EoS dataset was generated [23]. Furthermore, those two variables show small correlations with the input of our model, being less informative for the predictions. In a similar work in [44], the author also kept constant the

TABLE II. Extreme values and respective mean and standard deviation for nuclear matter parameters from the source dataset we are employing. The dataset contains a total of 25287 nuclear models. All quantities are in MeV except the saturation density n_0 , which is in fm^{-3} .

Y	\bar{Y}	σ_Y	Y_{\min}	Y_{\max}
n_0 (fm^{-3})	0.153351	0.003356	0.140047	0.166157
E_0	-16.099182	0.176263	-16.862894	-15.251849
K_0	269.555070	19.696655	193.038134	347.284418
Q_0	-381.870901	60.604233	-617.004518	-15.088353
J_{sym} (MeV)	31.870120	1.466915	25.416222	37.963163
L_{sym}	41.412327	9.824848	21.789640	93.972049
K_{sym}	-107.711394	31.334004	-212.745127	16.062865
Q_{sym}	1274.851388	297.082380	-63.828371	1962.652643

values of E_0 and n_0 , doing a study with the use of two NN architectures which first maps the observations to the EOS curve and then maps pressure times baryonic density to the parameters of nuclear matter.

The input X and output Y set are composed of \mathcal{D} rows of vectors \mathbf{x} and \mathbf{y} , respectively, where \mathcal{D} represents the dataset size we are utilizing. In simpler terms, Y is expressed as $Y = \{\mathbf{y}^{(i)}\}_{i=1}^{\mathcal{D}}$, and X is denoted as $X = \{\mathbf{x}^{(i)}\}_{i=1}^{\mathcal{D}}$. The size of the output vector is $\mathbf{y}^{(i)} = \{K_0, Q_0, J_{\text{sym}}, L_{\text{sym}}, K_{\text{sym}}, Q_{\text{sym}}\}$, while the input vector $\mathbf{x}^{(i)}$ is defined by 10 values; $\mathbf{x}^{(i)} = \{(M_j^{(i)}, R_j^{(i)})\}_{j=1}^5$, representing five simulated observations $M_j(R_j)$ in the mass-radius curve. Concerning the dataset size, we randomly divided the total EOS models into two distinct sets: a training set denoted as X and Y , incorporating 80% of the data ($\mathcal{D} = 22758$ EOS), and a test set denoted as X_T and Y_T , comprising the remaining 20% ($\mathcal{D}_T = 2529$ EOS). In summary, our output elements are six-dimensional vectors denoted as $\mathbf{y}^{(i)}$ with a total of training and test data expressed as $Y = \{\mathbf{y}^{(i)}\}_{i=1}^{22758}$ and $Y_T = \{\mathbf{y}^{(i)}\}_{i=1}^{2529}$, respectively. Regarding the input space, we have a ten-dimensional vector $\mathbf{x}^{(i)} = \{(M_j^{(i)}, R_j^{(i)})\}_{j=1}^5$, and both the training and test data are denoted as $X = \{\mathbf{x}^{(i)}\}_{i=1}^{22758}$ and $X_T = \{\mathbf{x}^{(i)}\}_{i=1}^{2529}$.

C. Observation mock data

The statistical procedure for generating the five mock observations $M_j(R_j)$ in the mass-radius curve with distinct input noises originating in different datasets unfolds through the following steps. For each EOS:

- (1) We randomly sample five NS mass values, $M_j^{(0)}$, from a uniform distribution within the range of $1.0M_\odot$ to M_{\max} ;
- (2) We sample five values, $\sigma_{j,M}$, from a uniform distribution spanning the interval $[0, \sigma_M]$ (σ_M is in Table III);
- (3) We sample five values, $\sigma_{j,R}$, from a uniform distribution spanning the interval $[0, \sigma_R]$ (σ_R is in Table III);

- (4) The NS radius, R_j , is sampled from a Gaussian distribution centered at the TOV solution, denoted as $R(M_j^{(0)})$, with the standard deviations $\sigma_{j,R}$;
- (5) Lastly, the NS mass is sampled from a Gaussian distribution centered at $M_j^{(0)}$ with the standard deviations $\sigma_{j,M}$.

This process is summarized by the following equations:

$$M_j^{(0)} \sim \mathcal{U}[1, M_{\max}] \quad (\text{in units of } M_\odot), \quad (10)$$

$$R_j \sim \mathcal{N}(R(M_j^{(0)}), \sigma_{j,R}^2), \quad \text{where } \sigma_{j,R} \sim \mathcal{U}[0, \sigma_R], \quad (11)$$

$$M_j \sim \mathcal{N}(M_j^{(0)}, \sigma_{j,M}^2), \quad \text{where } \sigma_{j,M} \sim \mathcal{U}[0, \sigma_M] \quad j = 1, \dots, 5. \quad (12)$$

This method is very similar to the method implemented in [38]. By performing the above numerical procedure, we construct the initial input structure $\mathbf{x} = [M_1, \dots, M_5, R_1, \dots, R_5]$, where each pair (M_j, R_j) is a single realization of the above procedure (*observation*). These pairs collectively characterize the $M(R)$ diagram of a given EOS. We will replicate n_s times the aforementioned procedure for the same EOS, i.e., n_s is the number of times we resample the input vector, \mathbf{x} , for the same EOS. As a result, the dataset is organized as $\mathbb{X} = \{\mathbf{X}_i\}_{i=1}^{n_s}$ and $\mathbb{Y} = \{\mathbf{Y}_i\}_{i=1}^{n_s}$. This approach expands our dataset to a size of $\mathbb{D} = n_s \times \mathcal{D}$, where \mathcal{D} denotes the original dataset size for the number of EOSs. For example, by choosing $n_s = 120$ implies the repetition of these procedures 120 times for each EOS, resulting in $\mathbb{X} = \{\mathbf{X}_i\}_{i=1}^{120} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_{120}\}$. We have generated a total of three distinct datasets, whose properties are detailed

TABLE III. Generation parameters for each dataset.

Dataset	$\sigma_M (M_\odot)$	σ_R (km)
0	0	0
1	0.136	0.626
2	0.271	1.253

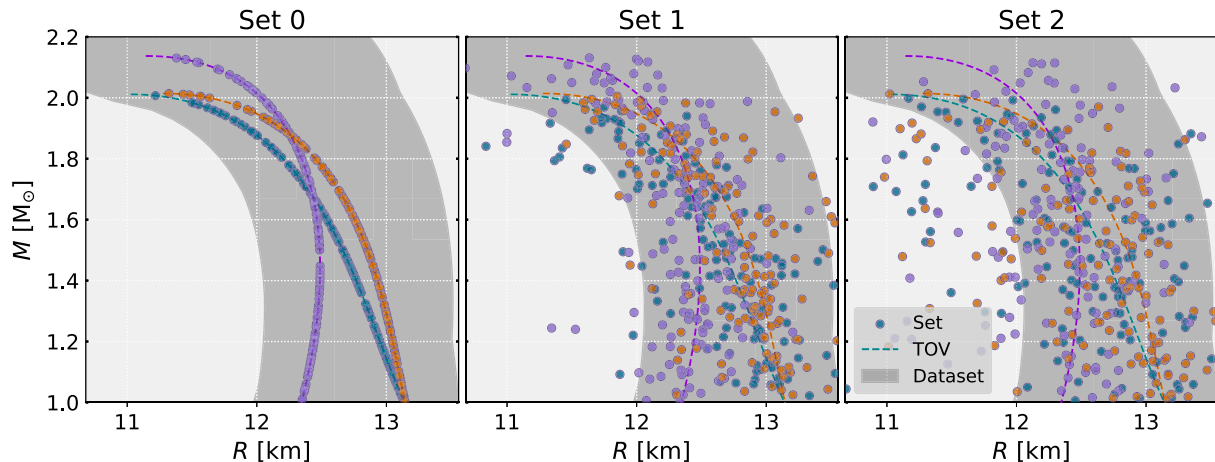


FIG. 2. The gray area represents the extremes of the dataset, while the three lines represent three different EOSs. The points along the lines represent the values corresponding to the respective sets, with the dataset size reduced to $n_s = 30$ for a clearer visualization.

in Table III. The standard deviation for set 2 was chosen by considering the ten initial values for the standard deviations of mass and radius in Table VII, which correspond to the ones with the smallest standard deviation for the radius. We calculated the mean of these ten values. Set 1 comprises half the values of set 2, and set 0 represents the values without noise, as illustrated in Fig. 2. The reason we only chose the ten values with a lower standard deviation of the radius is that otherwise, the input noise would be so high that the model would not be able to extract anything from it. The creation of these three sets aims to investigate how the model responds to different levels of noise. We utilized 120 mock observations ($n_s = 120$) in the training sets for each EOS, while $n_s = 1$ was employed for the test sets. This distinction mimics a real-world scenario where access to a single mock observation of the “true” EOS is typical. Here, a single mock observation corresponds to $n_s = 1$, representing five $M_j(R_j)$ mock observations. It’s essential to note that, for each EOS, there are 600 points in the $M(R)$ diagram, comprising 120 simulated observations for five neutron star mock observations each.

In our previous investigation [35], we employed $n_s = 60$ during the training phase. However, for the current study, where we have reduced the dimension of the output vector, we wanted to enhance the model’s performance without compromising training time and RAM usage. After experimenting with different adjustments to either the input vector or the n_s value for the training, we found that setting $n_s = 120$ yielded the optimal balance between improved performance and computational efficiency.

D. Training procedure

To investigate the response of BNNs to varying input noises and output targets, we conducted a series of experiments involving the training of diverse functional and stochastic models. The BNN models were trained

using distinct datasets generated following the guidelines provided in Sec. III C. In the training stage, a portion of the training data was randomly chosen as a validation set, with 80% dedicated to actual training and 20% for validation. Additionally, both the input data X and output data Y were standardized,¹ given the significant differences in the ranges of the output vector values. Defining the functional models entailed adjusting the number of neurons, layers, and activation functions. Table IV shows the best functional model for all the datasets mentioned in Table III.

In our exploration of hidden layers, we considered hyperbolic tangent, softplus, and sigmoid activation functions, with a linear activation function chosen for the output layer. The architecture incorporates three hidden layers, each consisting of ten neurons. The output layer consistently contains 12 neurons, with six dedicated to representing the mean and another six for the standard deviation of the output probability distribution function. It is noteworthy that we intentionally avoided incorporating correlation in the output layer, as its inclusion led to inferior performance. Consequently, the output layer is exclusively focused on capturing the mean and standard deviation information of the output distribution. The architecture adopted in this study features two to three hidden layers, maintaining a consistent number of neurons within each hidden layer.

The most favorable outcomes are achieved by employing the sigmoid activation function in the hidden layers, ensuring minimal loss and preventing divergence. It was determined that the optimal number of hidden layers is three. Detailed information on these configurations is available in Table IV. During training, we use a learning rate of 0.001 and employ the ADAM optimizer [58] with the AMSgrad improvement [59]. The models undergo training for 4000 epochs, utilizing a minibatch size of 1536.

¹ $z = (x - \mu)/\sigma$, where μ stands for the mean of the training dataset and σ is the standard deviation of the training dataset.

TABLE IV. Structures of the final BNN models.

Layers	Activation	Neurons
Input	N/A	10
Hidden layer 1	Sigmoid	10
Hidden layer 2	Sigmoid	10
Hidden layer 3	Sigmoid	10
Output	Linear	12

Concerning the stochastic model, we employ a Gaussian prior with a mean of zero and a standard deviation of one. While this prior choice lacks a specific theoretical justification, it serves as a reasonable default prior, as discussed in [48]. Future research could delve further into investigating the impact of prior parameters, similar to the approach taken in Ref. [60]. Additionally, we select a multivariate normal distribution as the variational posterior, initialized with a mean of 0 and a diagonal covariance matrix, where the standard deviation is equal to $\log(1 + \exp 0) = 0.693$. Furthermore, all BNNs models were coded using the Tensorflow library [61], more specifically we use Keras [62], an high-level API of the TensorFlow.

IV. RESULTS

After training our model, we proceed to assess its performance through a series of tests. Initially, we evaluate the model's accuracy using the test set derived from the original set of EOS outlined in Sec. IV A. Subsequently, in Sec. IV B, we examine the model's performance with different nuclear models. Lastly, we conduct a test on our model using the observations outlined in Sec. IV C. We should clarify that whenever we say, for instance, the results of set 1, we are specifically indicating the BNN model trained on dataset 1 and evaluated with a corresponding noise level from set 1. This applies to Secs. IV A and IV B. However, for the last Sec. IV C, it's crucial to highlight that no noise is applied to the test set, i.e., the test set consists of the mean values of the pulsar observations.

A. Test set prediction

Analyzing the predictions of our BNN model across the three sets, we begin by examining the behavior of model predictions over the entire test set. We define the normalized residuals as $\Gamma(Y) = (\hat{\mu}_Y - Y_T)/\hat{\sigma}_Y$, where $\hat{\mu}_Y$ and $\hat{\sigma}_Y$ are the mean and standard deviation predicted by the model, and Y_T is the true value. Subsequently, we evaluate the coverage probability.

The results for these two metrics are illustrated in Fig. 3 for our six-dimensional output vector across the three sets. Focusing on Γ (upper panel), we observe that for all three datasets, approximately 50% of the values cluster around zero. Additionally, the 95% confidence level is situated between $\pm 2\sigma$, with slight deviation for Q_{sym} . The analyses of the coverage probability (bottom panel) shows the

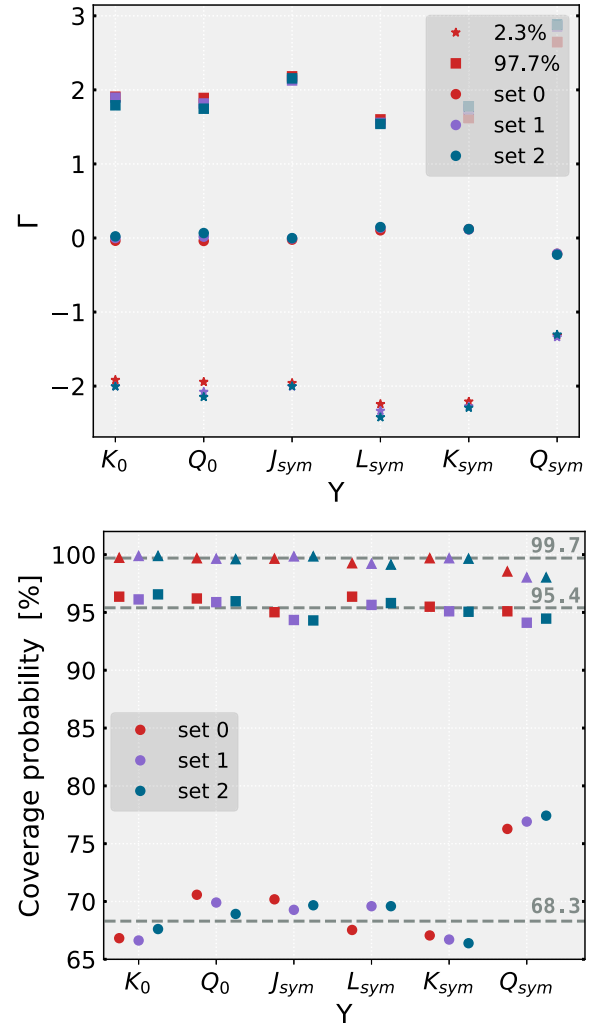


FIG. 3. In the upper plot, the median for $\Gamma(Y) = (\hat{\mu}_Y - Y_T)/\hat{\sigma}_Y$ is denoted by circular markers for each set, while the 95% confidence level is indicated between square markers (97.7% CL) and star markers (2.3% CL). In the bottom plot, the coverage probability is represented with circular markers for 68.3%, square markers for 95.4%, and triangular markers for 99.7% confidence levels.

model's ability to capture the data distribution. This metric quantifies the percentage of values within specific intervals relative to the total values in the test set. In particular, we examine whether the proportion of values falling within 1σ of the output distribution aligns with the expected 68.3%. This evaluation is then extended to 2σ (95.4%) and 3σ (99.7%) intervals. The model demonstrates an effective prediction of the associated percentage for the standard deviation being measured.

The observed behavior of Q_{sym} in the coverage probability can be attributed to the considerable skewness in the distribution of Γ , distinguishing it from the other nuclear properties.

It is crucial to note from these two plots that the results for the three sets are remarkably consistent. This observation

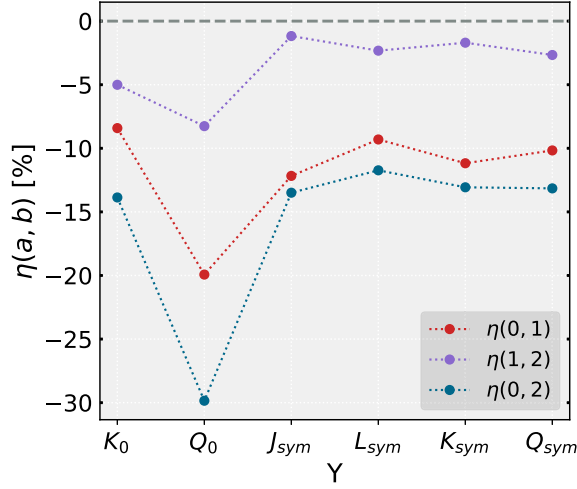


FIG. 4. Prediction uncertainty deviation $\eta(a, b)$ for the output vector for models a and b as defined in Eq. (13).

indicates that irrespective of the increased noise in our input vector, which leads to an increased uncertainty as evident in subsequent plots, the model consistently adapts to preserve the statistics of the dataset.

To gain a deeper insight into the impact of each set on the model prediction, we introduce the following metric:

$$\eta[a, b] = \frac{1}{T} \left(\sum_{i=1}^T \frac{\hat{\sigma}_i^a - \hat{\sigma}_i^b}{\hat{\sigma}_i^a} \right) \times 100. \quad (13)$$

This metric provides insights into the percentage uncertainty deviation between models trained with sets 'a' and 'b' across the six values of the output vector, and T is the total number of EOS in the test set. The results are shown in Fig. 4. Upon examining the plot, a notable observation is that, for all three η vectors, the quantity Q_0 demonstrates the most significant variation.

Qualitatively this aligns with our expectations, given that Q_0 exhibits the highest correlation with the observables, as illustrated in Fig. 12 in Appendix A. It's crucial to acknowledge that the correlation observed is only based on linear dependence. An alternative correlation measure, known as Kendall rank, is shown in Fig. 13 of the

work [23], where once again, Q_0 stands out with the highest score for $R_{1,4}$ and R_{\max} . Additionally, we acknowledge the inherent challenge posed by the stochastic nature of our model, which contributes to the complexity of quantitatively interpreting the results.

Moreover, the values of $\eta(0, 1)$, $\eta(1, 2)$, and $\eta(0, 2)$ consistently exhibit negative values, as anticipated. This is attributed to the absence of input noise in set 0 and the larger noise in the input of set 2 compared to set 1. Examining the order of η values, the blue curve exhibits a more negative trend, as expected, indicating a greater difference in input noise.

Additionally, $\eta(0, 1)$ appears more negative than $\eta(1, 2)$, emphasizing the impact of varying the input noise from a uniform distribution for sets 1 and 2. Occasionally, these two sets converge to similar standard deviation values for the noise, a scenario less likely for set 0, which consistently lacks input noise.

In the final examination of this section, we analyze the model's residuals, $\delta = (\hat{\mu}_Y - Y_T)$, by examining their mean and standard deviation, as detailed in Table V. Notably, the model exhibits higher accuracy for properties of nuclear matter with lower order, which is correlated with their smaller range of values. When comparing between sets, it becomes apparent that the standard deviation of residuals is greater for set 2, followed by set 1, and, finally, set 0. This observation indicates that the model achieves higher accuracy when trained with less input noise, aligning with our expectations.

For instance, the L_{sym} residuals standard deviation (MeV) for the model predictions is 8.77 (set 0), 9.52 (set 1), 9.69 (set 2). These uncertainties are not larger, and generally smaller than the ones obtained in several works where the uncertainties linked to experimental, observational or theoretical uncertainties are of the order 10–28% [63–65].

B. Nuclear models application

After confirming the ability of our model to predict the test set, we extend our assessment to evaluate its performance with additional nuclear models. Specifically, we consider a set of 31 unified EOS built from a relativistic

TABLE V. The mean and the standard deviation of the model residuals $\delta = (\hat{\mu}_Y - Y_T)$ for the three sets.

Y (MeV)	Set 0		Set 1		Set 2	
	$\bar{\delta}$	σ_δ	$\bar{\delta}$	σ_δ	$\bar{\delta}$	σ_δ
K_0	-0.663391	16.440592	-0.505939	17.894562	-0.315610	18.705447
Q_0	-1.879105	39.562439	-1.338307	48.660290	-0.381916	52.664798
J_{sym}	-0.002277	1.272795	0.003521	1.436066	0.003998	1.454013
L_{sym}	0.004316	8.769408	0.039834	9.523284	0.021291	9.692659
K_{sym}	0.157757	27.521557	-0.311028	30.918154	-0.179070	31.494850
Q_{sym}	2.541514	280.488010	2.908357	291.857966	2.560507	293.681477

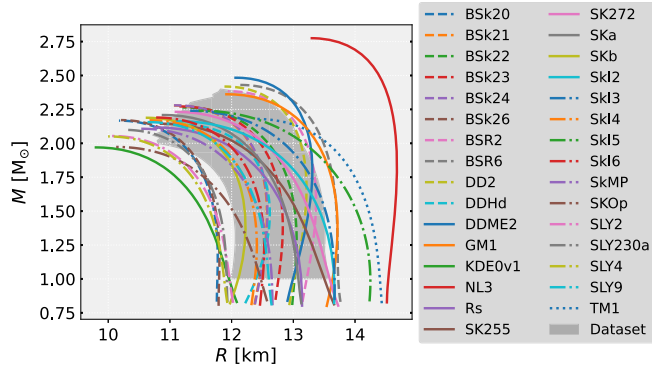


FIG. 5. The gray band represents the dataset region without noise, and the multiple lines represent the 31 nuclear models. We have a total of 14 EOS inside the dataset region without noise for the input space and 12 EOS inside the dataset region for the output space.

mean field (RMF) approach and nonrelativistic Skyrme interactions [66]. These EOS have been considered because they are unified, meaning that the inner crust and the core have been obtained from the same model, and they span a large set of nuclear matter properties. This last characteristic will allow us also to test the extrapolation ability of our BNN model. Besides, this set has already been used in a previous study using ML methods [36], and, therefore some comparisons are possible. The mass-radius curves for these models are represented in Fig. 5.

In Sec. IV A we have measured the performance of three BNN models, each one trained in a distinct dataset (see Table III). Here, we are applying these models to an independent set of nuclear models. For that, we first need to generate the corresponding observation sets (input data) with the same statistical properties as the ones in which the

models were trained. For instance, if we want to apply the BNN model trained on set 0, we need to generate an observation set for these nuclear models without any noise (see Table III). Furthermore, the generation of the observation sets follows a similar procedure used in Sec. III C, albeit with some modifications. For each nuclear model and level noise, the procedure is as follows: (i) randomly selecting 20 points from each $M(R)$ curve, (ii) from these 20 points, 100 sets of five points are generated by sampling without replacement, resulting in 100 distinct input vectors \mathbf{x} with 10 values each.

The selection of 20 points for the mass-radius $M(R)$ curve was driven by the number of available observations, as indicated in Table VII, which is 18. Given the relatively small size of the input vector, consisting of only five pairs, we opted to generate the 100 subsets for each nuclear model.

In order to analyze the predictions obtained with the BNN models trained in the three datasets for each nuclear model, we represent the BNN model's performance (colored dots with error bars) and the real value of the nuclear model defined as NM_k (black dots), where k represents each of the 31 nuclear models in Figs. 6 and 7.

The predictions for the 100 samples of each nuclear model are presented utilizing both the law of total expectation and the law of total variance, with mean and standard deviation values denoted as $\hat{\mu}_k$ and $\hat{\sigma}_k$, respectively, where once again k represents each of the 31 nuclear models. These quantities are represented by an error bar defined as $\hat{\mu}_k \pm 2\hat{\sigma}_k$ for the three datasets. We include the mean (\bar{Y}) (dashed line) and two standard deviations ($2\sigma_Y$) of the training dataset in these figures (dark gray band). Additionally, the range of the training dataset [Y_{\min} , Y_{\max}] is shown (light gray band), providing a reference

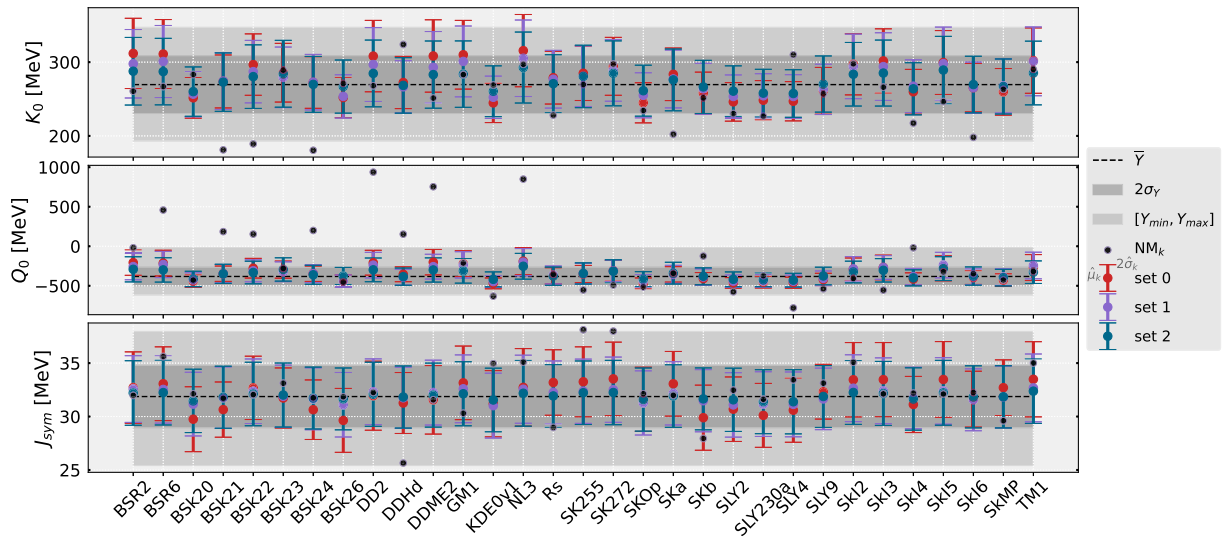


FIG. 6. Representation of the K_0 , Q_0 , J_{sym} output values, where the light gray band illustrates the minimum and maximum values within the training dataset [Y_{\min} , Y_{\max}]. The dashed line and dark gray band correspond, respectively, to the mean and the 2σ CI of the training region ($\bar{Y} \pm 2\sigma_Y$). The error bar denotes the 2σ range prediction by the BNN model ($\hat{\mu}_k \pm 2\hat{\sigma}_k$) for the sets 0, 1, and 2.

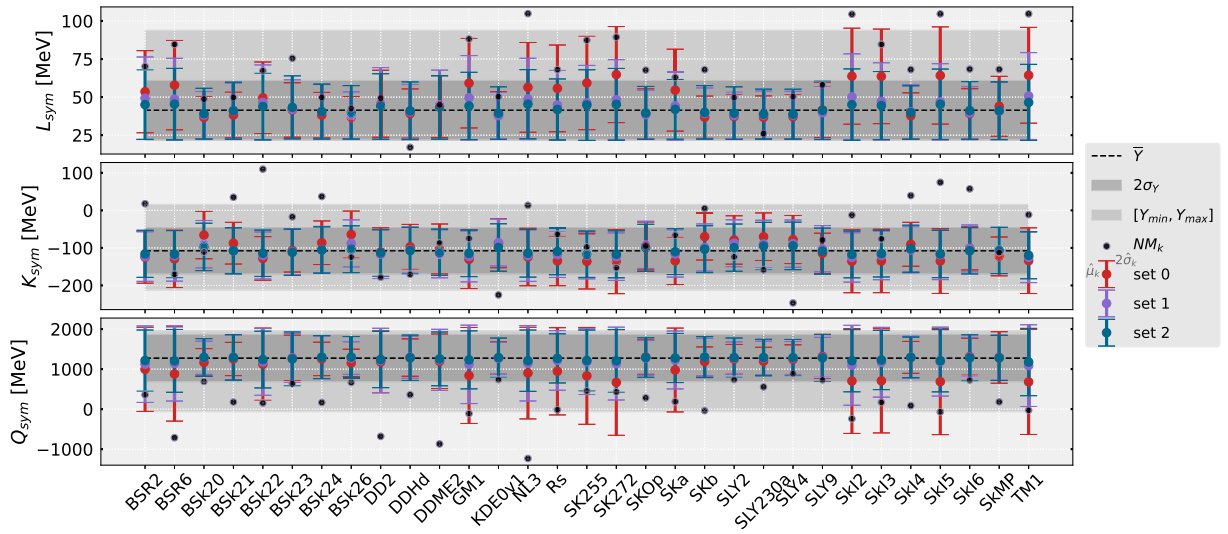


FIG. 7. Representation of the L_{sym} , K_{sym} , Q_{sym} output values, where the light gray band illustrates the minimum and maximum values within the training dataset $[Y_{\text{min}}, Y_{\text{max}}]$. The dashed line and dark gray band correspond, respectively, to the mean and the 2σ CI of the training region ($\bar{Y} \pm 2\sigma_Y$). The error bar denotes the 2σ range prediction by the BNN model ($\hat{\mu}_k \pm 2\hat{\sigma}_k$) for the sets 0, 1, and 2. More information can be found in the text.

for understanding the BNN model's performance relative to the training distribution. For the higher-order properties, particularly for Q_0 , the BNN model struggles when values significantly deviate from the training region, becoming unable to accurately track the true value. Notably, models trained with the set without uncertainty show an ability to appropriately extend the error bar, effectively capturing the true value. For certain properties, such as Q_{sym} , these models even reach beyond the training region. In general, it is expected that if the model finds useful physical information in the input data, $\hat{\sigma}_k < \sigma_Y$ (the model is confident because it has seen similar data during training). On the other hand, if there is no useful physical information, i.e., too much noise, or the nature of the input data differs significantly from the training data set, i.e., model sees unconventional input data with respect to the training data, then the model will increase the uncertainty prediction to fit the true value within its uncertainty band prediction, i.e., $\hat{\sigma}_k > \sigma_Y$. Whenever the NM_k data are outliers (Figs. 6 and 7), it is the red error bars associated with set 0 (no noise) that fall closer to the NM_k ; when the NM_k falls within the training region, it is mostly contained within the red error bars. This is only true if the noise introduced into the input data is sufficiently small. A correlation study between the BNN mean predictions and the nuclear models values is presented in Appendix A for completeness.

To gain a clearer understanding of the BNN model's predictive performance, we have recalculated the coverage probability for 1σ , 2σ , and 3σ , as illustrated in Fig. 8 for these new three datasets. This provides insight into the percentage of values our BNN model accurately captures up to 3σ . Notably, J_{sym} exhibits nearly 100% coverage for 3σ , indicating that the model effectively contains the

majority of values within a 3σ range from the predicted mean.

When comparing the inference results between the test set of Sec. IVA (see Fig. 3, right panel) and the above figure for the nuclear models (Fig. 8), it is clear a considerable decline in the BNN models performances. However, we should consider the key difference between both inferences; while the test set of Sec. IVA has exactly the same statistical structure of the training set, these 31 nuclear models are characterized by nuclear matter properties values far from the training region; in other words, the above inferences heavily rely on the extrapolation capacity

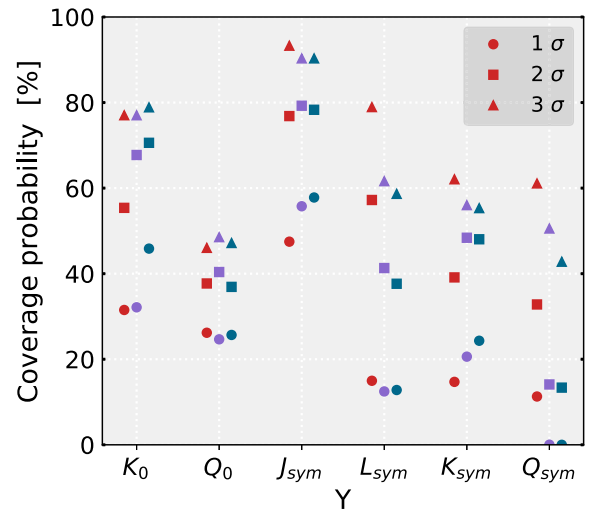


FIG. 8. Coverage probability for the model tested with the three sets across 100 samples of the 31 nuclear models, with red representing set 0, purple representing set 1, and blue representing set 2.

of BNN models. One obvious way of increasing the BNN models performance would be to generalize the training set to be as broad as possible, covering an extensive range of possible nuclear matter properties. However, the training set we have used was restricted to be consistent with theoretical/experimental low-energy constraints: in other words, we are not expecting the true EOS of nuclear matter, and its nuclear matter parameters, to strongly deviate from our training statistics. Notice that the nuclear models that fail to reproduce with good accuracy the nuclear matter properties are distributed within the different frameworks used to build the EOS, i.e., the performance depends on the way the phenomenological nuclear models are constrained and not on the different frameworks, such as RMF, Skyrme forces, or hadronic density dependent models.

The performance of the BNN models is well captured by the two combined heatmaps represented in Fig. 9 illustrating a comparison between the training dataset properties Y and the BNN model predictions $\hat{\mu}_k$ for the 31 nuclear models, with the true nuclear matter properties of these 31 EOS, respectively, in the top and bottom heatmap.

In the upper panel, we visualize the deviation of target values, specifically the six nuclear matter properties denoted as NM_k for nuclear model k , from the mean of the training region, \bar{Y} . This deviation is normalized by the standard deviation of the training region, σ_Y . Meanwhile, the lower panel portrays the absolute value of the normalized model residuals' predictions for set 2, chosen as an illustrative example. To be precise, this representation

captures the distance between the true value NM_k and the predicted mean for each nuclear model, $\hat{\mu}_k$, normalized by the predicted standard deviation, $\hat{\sigma}_k$. These individual quantities were shown in Figs. 6 and 7. The objective of this visualization is to emphasize that nuclear model properties with a greater distance from the mean of the training data also tend to exhibit a larger deviation from the BNN predicted mean, $\hat{\mu}_k$. This trend is evident, for instance, in the case of the quantity Q_0 . Taking the nuclear model DD2 as an illustrative example, Q_0 exhibits the furthest deviation from the mean value of the training distribution and, consequently, is the farthest from the predicted mean being $15 \hat{\sigma}_{DD2}$ away from the predicted mean. The ranges of the mean values and standard deviations in the training set vary significantly for each quantity, as detailed in Table II.

In a recent study [44], a deep learning model was also tested with nuclear models, albeit with a different NM amount and output/input quantities. Notably, we utilized two common models, SK272 and SK255, specifically for the properties L_{sym} , K_{sym} , and K_0 . Remarkably, we achieved smaller residuals in comparison to their results for the quantities K_{sym} and K_0 , despite the fact that they employed a more complex architecture and conducted a direct mapping from the EOS to the properties of nuclear matter.

Furthermore, in our earlier work [36], where we directly mapped the EOS of β -stable matter to the properties of nuclear matter, we also conducted tests with the same 31 nuclear models. The results were not only successful but also exhibited a higher precision. However, our main

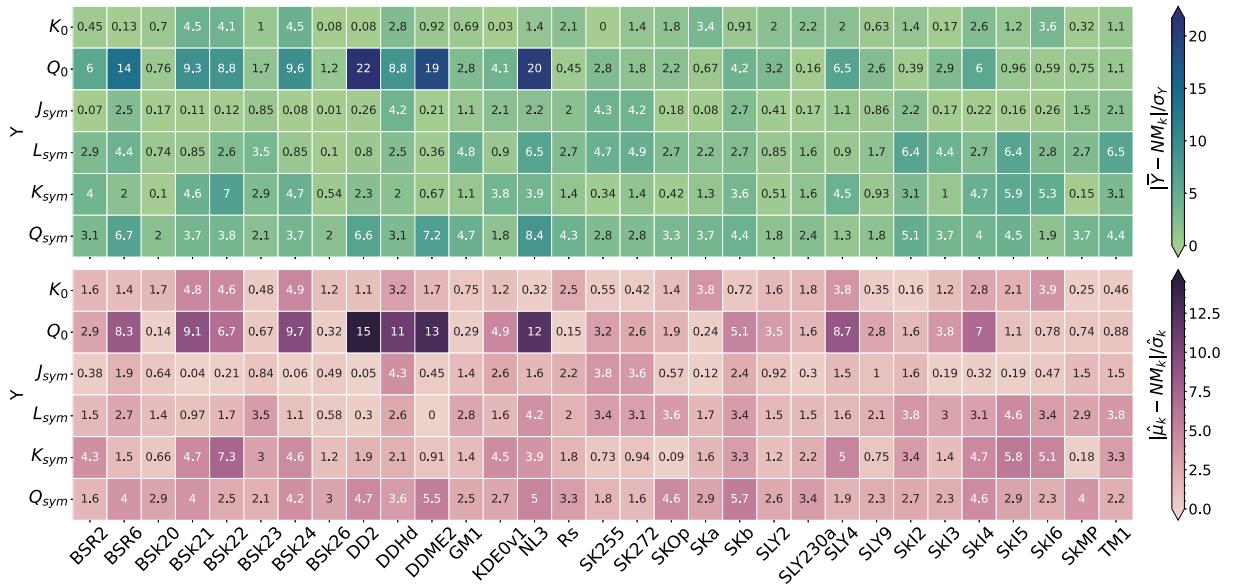


FIG. 9. In the top panel of the heatmap, we present an overview of the BNN model predictions' reliability, considering the normalized distance of target values (nuclear matter properties NM_k for each model k) from the respective mean of the training region, $|\bar{Y} - NM_k|/\sigma_Y$. Here, \bar{Y} and σ_Y denote the mean and standard deviation, respectively, of the training dataset. The values NM_k correspond to the nuclear matter properties, where the index k represents each nuclear model. In the bottom panel, the focus shifts to the absolute value of the normalized model residuals' predictions for set 2, and the training dataset mean and standard deviation are substituted by $\hat{\mu}_k$ and $\hat{\sigma}_k$ that symbolize, respectively, the predicted mean and standard deviation, $|\hat{\mu}_k - NM_k|/\hat{\sigma}_k$. For a detailed understanding of these individual quantities, see Figs. 6 and 7.

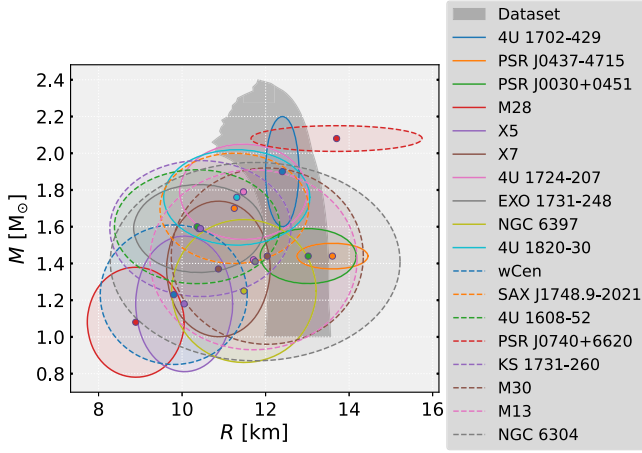


FIG. 10. Observations that come from the Table VII, for the 68% CL, the dark gray band is the dataset region without noise. Notice that there are only three mean values inside the training region.

drawback in that study was the challenge of quantifying the uncertainty, a challenge that we have effectively addressed in our present work.

C. Real observation dataset

In a final evaluation of our BNN model, we conducted a final test using the real NS observations listed in Table VII. Figure 10 shows these observations. As their dispersion is quite high and most of them are far from the training set, we took their mean values as the observation values, which we supplied to the BNN model.

Contrarily to the last section, here we employ just one BNN model to the pulsar dataset. We use a BNN model trained on set 2 because the noise properties of set 2 were determined precisely from the pulsar list, Table VII, leading to set 2 exhibiting an input noise similar to the standard deviation of the mass-radius pairs. Additionally, this choice is motivated by the highly dispersed nature of the mean values of the pulsar observations, which fall within the

TABLE VI. The predicted mean and standard deviation, denoted as $\hat{\mu}_{\text{obs}}$ and $\hat{\sigma}_{\text{obs}}$ respectively, determined for the output vector based on the set created from the observations listed in Table VII. Further details can be found in the accompanying text.

Y (MeV)	Set 2	
	$\hat{\mu}_{\text{obs}}$	$\hat{\sigma}_{\text{obs}}$
K_0	264.043724	18.184857
Q_0	-397.115823	55.868526
J_{sym}	31.622006	1.466793
L_{sym}	39.802926	8.764159
K_{sym}	-101.667751	31.432478
Q_{sym}	1294.825496	252.531340

domain of set 2. This implies that the model has learned these diverse regions effectively. In a similar methodology as followed in the last section, we generate sets of five pairs from the 18 pulsar values through sampling without replacement, and this process was iterated 100 times, resulting in a dataset of 100 input vectors for testing. We opted for 100 repetitions to remain consistent with the previous subsection.

Only three mean values fall within the dataset region without noise for the mass radius curves, and the dispersion of each observation, in most cases, surpasses the radius interval covered during training. This implies that the model is extrapolating values, making it challenging to obtain accurate predictions. The resulting mean and standard deviation for the six values of the output vector were obtained using, once again, the law of total variance and the law of total expectation for the 100 different input vectors. These quantities are represented as $\hat{\mu}_{\text{obs}}$ and $\hat{\sigma}_{\text{obs}}$ respectively, and are presented in Table VI. To enhance predictions, more precise observation values are crucial.

The comparison between our work and existing studies for the quantities of symmetry energy, its slope, and curvature is presented in Fig. 11. Specifically, we compare

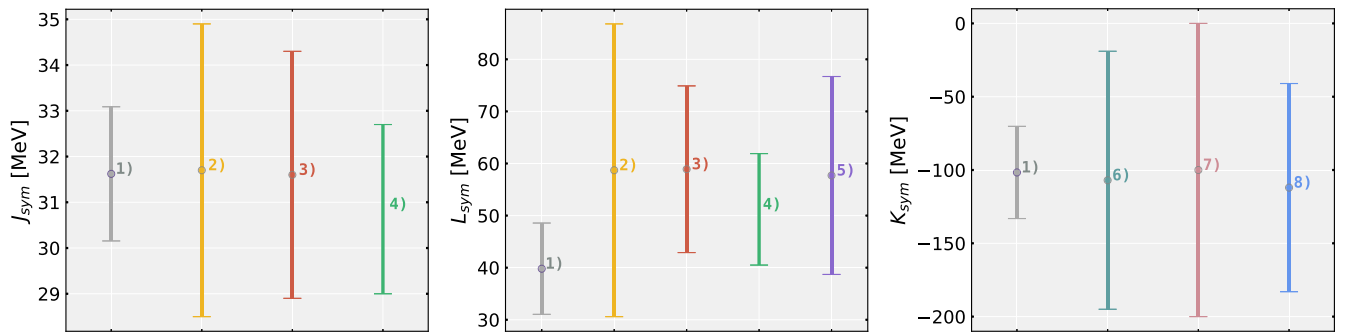


FIG. 11. Comparison of our results for the symmetry energy, its slope, and curvature (labeled as 1 for a 68% CL) with values from other studies; 2) $J_{\text{sym}} = 31.7 \pm 3.2$ MeV and $L_{\text{sym}} = 58.7 \pm 28.1$ MeV from [65]; 3) $J_{\text{sym}} = 31.6 \pm 2.7$ MeV and $L_{\text{sym}} = 58.9 \pm 16$ MeV [68]; 4) $29 < J_{\text{sym}} < 32.7$ MeV and $40.5 < L_{\text{sym}} < 61.9$ MeV [64]; 5) $L_{\text{sym}} = 57.7 \pm 19$ MeV at a 68% confidence level [69]; 6) $K_{\text{sym}} = -107 \pm 88$ MeV at a 68% confidence level in [69]; 7) $K_{\text{sym}} = -100 \pm 100$ MeV [70]; and 8) $K_{\text{sym}} = -112 \pm 71$ MeV [71].

our results: 1) for a 68% CL with the following references; 2) $J_{\text{sym}} = 31.7 \pm 3.2$ MeV and $L_{\text{sym}} = 58.7 \pm 28.1$ MeV from [65]; 3) $J_{\text{sym}} = 31.6 \pm 2.7$ MeV and $L_{\text{sym}} = 58.9 \pm 16$ MeV [68]; 4) $29 < J_{\text{sym}} < 32.7$ MeV and $40.5 < L_{\text{sym}} < 61.9$ MeV [64]; 5) $L_{\text{sym}} = 57.7 \pm 19$ MeV at a 68% confidence level [69] for 24 new analysis of neutron star data since GW170817; 6) $K_{\text{sym}} = -107 \pm 88$ MeV at a 68% confidence level in [69] but for the 16 new analysis of neutron star data since GW170817; 7) $K_{\text{sym}} = -100 \pm 100$ MeV [70]; and 8) $K_{\text{sym}} = -112 \pm 71$ MeV [71]. The key takeaway from this comparison is that our results for the three quantities fall within the ranges reported by the seven different works, suggesting the accuracy of our model. Particularly noteworthy is the alignment of our predicted mean values with those from other studies, especially for J_{sym} and K_{sym} .

V. CONCLUSIONS

Throughout this study, we have delved into the exploration of nuclear properties, aiming to derive them directly from mass-radius observations while quantifying their respective uncertainties. Our approach involves the application of BNNs, a probabilistic machine learning model that outperforms traditional neural networks by providing uncertainty measurements alongside its predictions. Our EOS dataset used to train the BNN model was built from a relativistic mean field approach within a Bayesian framework and incorporates crucial constraints derived from both nuclear matter properties and neutron star observations. Unlike more flexible EOS parametrizations discussed in works like [32], we deliberately opted for this specific family of microscopic nuclear models. This choice is driven by our overarching goal—to explore the feasibility of inferring nuclear matter properties, such as the quantities $K_0, Q_0, J_{\text{sym}}, L_{\text{sym}}, K_{\text{sym}}, Q_{\text{sym}}$, from observations of neutron stars. Spanning a dataset of 25 287 EOS—we generated three distinct sets of mock observational data for the mass and radius of five NS, simulating diverse scenarios of uncertainties, as outlined in Table III. These sets represent various levels of input noise scatter. Subsequently, we trained three BNN models, each corresponding to one of the three sets.

Through this exploration, we have successfully demonstrated the efficacy of BNNs in utilizing measurements of the mass and radius from five neutron stars to extract comprehensive information about the properties of nuclear matter. Our model has demonstrated its capability to provide valuable insights while effectively addressing uncertainties, particularly for the test set crafted with the same statistical characteristics as the training set.

We further extended our investigation by testing the BNN model using the NS EOS derived from 31 nuclear models [66], probing its ability to predict the properties of nuclear matter when confronted with samples exhibiting a

behavior vastly different from the training set. Some of these samples could fall completely outside the training region, as illustrated in Fig. 5. Adapting the 31 NS EOS to the three distinct sets with different uncertainties, we observed that the BNN model encountered challenges when dealing with samples that fall far away from the mean of the training region. Interestingly, we found that set 0, which included no uncertainty on the measurement of the mock data, demonstrated a superior ability to accurately capture the target values beyond the training region, even extrapolating successfully, going towards the direction of the out of the training region values.

In a final test, we evaluated the model using a test set with real observation values, once again encountering values completely outside the training region and exhibiting a quite large dispersion. This test dataset was exclusively tested on the model trained with set 2, i.e., the set with the largest uncertainties on the mock data observations. Our key takeaway from this test is that the current available observations still require improvement, given the substantial associated uncertainty leading to limited information for the extraction of the EOS. We anticipate advancements from future observations, such as those expected from STROBE-X [16] and eXTP [14], promising radius measurements with uncertainties as low as 2%–5%, thereby refining our predictions.

Numerous approaches for further exploration and improvement unfold from our current work. One potential direction involves augmenting the output vector by incorporating fourth-order components of the Taylor expansion of the nuclear matter EOS.

One particularly promising idea for future work involves training the BNN model with a diverse range of EOS that permit the extraction of nuclear matter properties (e.g., excluding polytropics and other agnostic approaches). Specifically, this could involve training with the nuclear models tested in Sec. IV B and utilizing metamodels. The BNN model would be able to learn a larger domain for the input and output space.

Introducing hyperons or a quark phase into the composition of neutron stars and training and testing the model with this set could offer intriguing insights. Regarding the stochastic model, developing other prior distributions for the weights is worth for exploration. Additionally, a crucial study would involve analyzing other distributions beyond Gaussian distributions for the posterior of the weights. Exploring alternative distributions could potentially lead to further improvements in the loss optimization. These improvements collectively offer rich possibilities for advancing and refining our current methodology.

ACKNOWLEDGMENTS

This work was partially supported by national funds from FCT (Fundação para a Ciência e a Tecnologia, I.P, Portugal) under the Projects No. 2022.06460.PTDC

(with the DOI identifier 10.54499/2022.06460.PTDC) No. UIDB/04564/2020 and No. UIDP/04564/2020 (with DOI identifiers 10.54499/UIDB/04564/2020 and 10.54499/UIDP/04564/2020) respectively.

APPENDIX A: CORRELATIONS

Let us calculate and analyze the Pearson correlation, for the source dataset, between the parameters of nuclear matter and the corresponding radius values for fixed mass values. While it is a limited quantity as it only measures linear correlations between variables, it already gives a

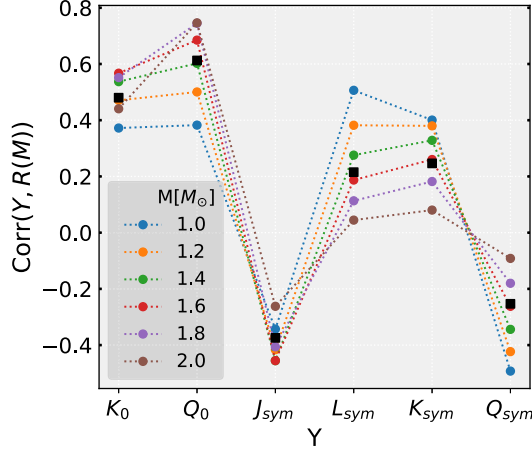


FIG. 12. Correlation between the nuclear matter parameters and the radius for fixed mass values $R(M)$. In black squares we show the mean correlation value for $M/M_\odot \in [1, 2]$.

good indication of the important properties in describing the NS radius. The results are shown in Fig. 12. The most notable observation is that the quantity Q_0 exhibits the highest correlation coefficient. By considering the isovector nuclear matter properties, $\{J_{\text{sym}}, \dots, Q_{\text{sym}}\}$, it is clear an interesting feature; the low-order parameters are more strongly correlated with the low M while the opposite happens for high order parameters. This is expected since low NS masses have central densities much smaller than massive NS and thus low-order parameters are enough to explain the radii of light NS. On the other hand, massive NS radii are more sensitive to high-order parameters because their central densities are very sensitive to the high-order polynomial orders in the EOS. This plot serves as a valuable tool for comprehending the results of this work. A deeper study on the subject of correlation can be found in [72].

In Fig. 13 we show the scatter plots $(\hat{\mu}_k, \text{NM}_k)$, i.e., the BNN predictions vs the nuclear models values, with the correspondent correlation coefficients for each of the three test sets. Lastly, the correlations among the nuclear matter properties for the training dataset and the nuclear models are displayed in Fig. 14. Several conclusions can be drawn: (i) the correlations between the two sets (Fig. 13) are generally positive and larger for set 0, although in some cases the correlations are very small or the difference between the sets is small; (ii) set 0 has the most extreme values of $\hat{\mu}_k$, the smallest and the highest, because this is the set that comes closer to the NM_k ; (iii) the negative correlation obtained for K_{sym} could be due to the correlations that the BNN model has learned from the training

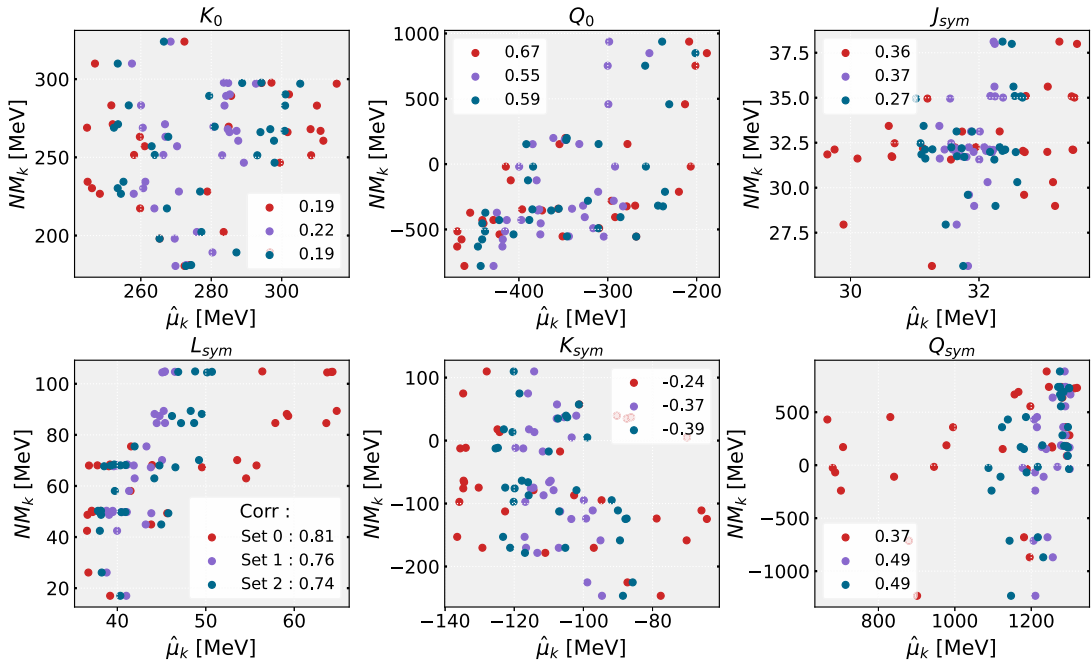


FIG. 13. Scatter plots of $(\hat{\mu}_k, \text{NM}_k)$ for each nuclear matter parameter and the corresponding correlation values between $\hat{\mu}_k$ and NM_k for each set.

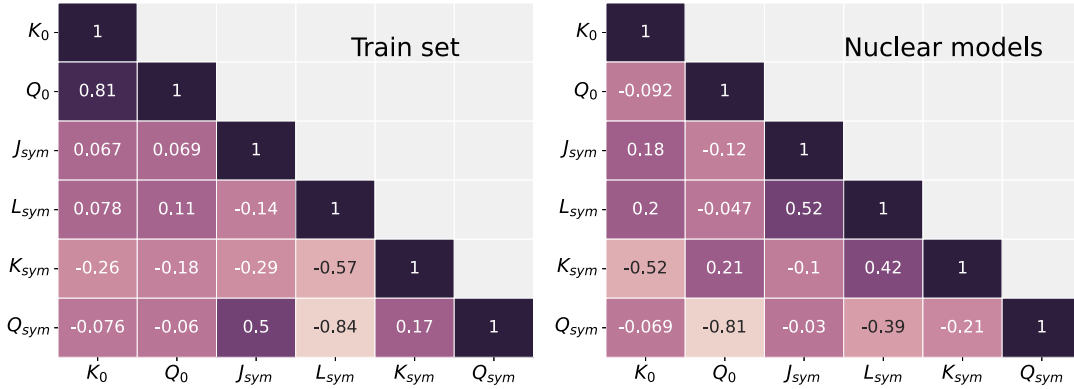


FIG. 14. The correlations between the nuclear matter properties for the training dataset (left) and for the 31 nuclear models (right).

set, which shows a negative correlation of the order of -0.5 between L_{sym} and K_{sym} (Fig. 14). On the other hand, the 31 nuclear models show a positive correlation of the order of 0.4 between L_{sym} and K_{sym} . The BNN model has learned the correlations of the training set, but tries to overcome them in set 0, which has no noise, since set 0 has the least negative correlation, followed by set 1 and set 2 according to the degree of noise. The behavior of K_{sym} also affects the magnitude of the correlations of Q_{sym} .

APPENDIX B: TABLE OF OBSERVATIONS

Illustration of the observation table outlined in the article [67], featuring 18 observables derived through Gaussian-fitted analyses. These observations are established from marginalized distributions, capturing key properties associated with neutron stars.

TABLE VII. Observation table sourced from the article [67].

Observables	M (M_\odot)		R (Km)	
	Mean	σ	Mean	σ
4U 1702-429	1.9	0.3	12.4	0.4
PSR J0437-4715	1.44	0.07	13.6	0.85
PSR J0030+0451	1.44	0.15	13.02	1.15
M28	1.08	0.3	8.89	1.16
X5	1.18	0.37	10.05	1.16
X7	1.37	0.37	10.87	1.24
4U 1724-207	1.79	0.26	11.47	1.53
EXO 1731-248	1.59	0.24	10.4	1.56
NGC 6397	1.25	0.39	11.48	1.73
4U 1820-30	1.76	0.26	11.31	1.75
wCen	1.23	0.38	9.8	1.76
SAX J1748.9-2021	1.7	0.3	11.25	1.78
4U 1608-52	1.6	0.31	10.36	1.98
PSR J0740+6620	2.08	0.07	13.7	2.05
KS 1731-260	1.59	0.37	10.44	2.17
M30	1.44	0.48	12.04	2.3
M13	1.42	0.49	11.71	2.48
NGC 6304	1.41	0.54	11.75	3.47

- [1] P. Demorest, T. Pennucci, S. Ransom, M. Roberts, and J. Hessels, *Nature (London)* **467**, 1081 (2010).
- [2] E. Fonseca *et al.*, *Astrophys. J.* **832**, 167 (2016).
- [3] Z. Arzoumanian *et al.* (NANOGrav Collaboration), *Astrophys. J. Suppl. Ser.* **235**, 37 (2018).
- [4] J. Antoniadis *et al.*, *Science* **340**, 448 (2013).
- [5] E. Fonseca *et al.*, *Astrophys. J. Lett.* **915**, L12 (2021).
- [6] R. W. Romani, D. Kandel, A. V. Filippenko, T. G. Brink, and W. Zheng, *Astrophys. J. Lett.* **908**, L46 (2021).
- [7] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Phys. Rev. X* **9**, 011001 (2019).
- [8] R. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), *Astrophys. J. Lett.* **896**, L44 (2020).
- [9] T. E. Riley, A. L. Watts, S. Bogdanov, P. S. Ray, R. M. Ludlam, S. Guillot, Z. Arzoumanian, C. L. Baker, A. V. Bilous, D. Chakrabarty, K. C. Gendreau, A. K. Harding, W. C. G. Ho, J. M. Lattimer, S. M. Morsink, and T. E. Strohmayer, *Astrophys. J.* **887**, L21 (2019).
- [10] M. C. Miller *et al.*, *Astrophys. J.* **887**, L24 (2019).
- [11] T. E. Riley, A. L. Watts, P. S. Ray, S. Bogdanov, S. Guillot, S. M. Morsink, A. V. Bilous, Z. Arzoumanian, D. Choudhury, J. S. Deneva *et al.*, *Astrophys. J. Lett.* **918**, L27 (2021).
- [12] M. C. Miller, F. Lamb, A. Dittmann, S. Bogdanov, Z. Arzoumanian, K. Gendreau, S. Guillot, W. Ho, J. Lattimer, M. Loewenstein *et al.*, *Astrophys. J. Lett.* **918**, L28 (2021).
- [13] G. Raaijmakers, S. Greif, K. Hebeler, T. Hinderer, a. Nissanke, A. Schwenk, T. Riley, A. Watts, J. Lattimer, and W. Ho, *Astrophys. J. Lett.* **918**, L29 (2021).
- [14] A. L. Watts *et al.*, *Sci. China Phys. Mech. Astron.* **62**, 29503 (2019).
- [15] S.-N. Zhang *et al.* (eXTP Collaboration), *Sci. China Phys. Mech. Astron.* **62**, 29502 (2019).
- [16] P. S. Ray *et al.* (STROBE-X Science Working Group Collaboration), [arXiv:1903.03035](https://arxiv.org/abs/1903.03035).
- [17] A. Watts *et al.*, *Proc. Sci.*, AASKA14 (2015) 043 [[arXiv:1501.00042](https://arxiv.org/abs/1501.00042)].
- [18] K. Hebeler, J. M. Lattimer, C. J. Pethick, and A. Schwenk, *Astrophys. J.* **773**, 11 (2013).
- [19] C. Drischler, K. Hebeler, and A. Schwenk, *Phys. Rev. Lett.* **122**, 042501 (2019).
- [20] J. Ghiglieri, A. Kurkela, M. Strickland, and A. Vuorinen, *Phys. Rep.* **880**, 1 (2020).
- [21] A. W. Steiner, J. M. Lattimer, and E. F. Brown, *Astrophys. J.* **722**, 33 (2010).
- [22] T. Malik, M. Ferreira, B. K. Agrawal, and C. Providência, *Astrophys. J.* **930**, 17 (2022).
- [23] T. Malik, M. Ferreira, M. B. Albino, and C. Providência, *Phys. Rev. D* **107**, 103018 (2023).
- [24] R. Essick, P. Landry, and D. E. Holz, *Phys. Rev. D* **101**, 063007 (2020).
- [25] P. Landry, R. Essick, and K. Chatzioannou, *Phys. Rev. D* **101**, 123007 (2020).
- [26] M.-Z. Han, J.-L. Jiang, S.-P. Tang, and Y.-Z. Fan, *Astrophys. J.* **919**, 11 (2021).
- [27] T. Gorda, O. Komoltsev, and A. Kurkela, *Astrophys. J.* **950**, 107 (2023).
- [28] L. Lindblom and N. M. Indik, *Phys. Rev. D* **86**, 084003 (2012).
- [29] A. Kurkela, E. S. Fraga, J. Schaffner-Bielich, and A. Vuorinen, *Astrophys. J.* **789**, 127 (2014).
- [30] E. R. Most, L. R. Weih, L. Rezzolla, and J. Schaffner-Bielich, *Phys. Rev. Lett.* **120**, 261103 (2018).
- [31] E. Lope Oter, A. Windisch, F. J. Llanes-Estrada, and M. Alford, *J. Phys. G* **46**, 084001 (2019).
- [32] E. Annala, T. Gorda, A. Kurkela, J. Nättilä, and A. Vuorinen, *Nat. Phys.* **16**, 907 (2020).
- [33] E. Annala, T. Gorda, E. Katerini, A. Kurkela, J. Nättilä, V. Paschalidis, and A. Vuorinen, *Phys. Rev. X* **12**, 011058 (2022).
- [34] K. Zhou, L. Wang, L.-G. Pang, and S. Shi, *Prog. Part. Nucl. Phys.* **135**, 104084 (2024).
- [35] V. Carvalho, M. Ferreira, T. Malik, and C. Providência, *Phys. Rev. D* **108**, 043031 (2023).
- [36] M. Ferreira, V. Carvalho, and C. Providência, *Phys. Rev. D* **106**, 103023 (2022).
- [37] M. Ferreira and C. Providência, *J. Cosmol. Astropart. Phys.* **07** (2021) 011.
- [38] Y. Fujimoto, K. Fukushima, and K. Murase, *J. High Energy Phys.* **03** (2021) 273.
- [39] Y. Fujimoto, K. Fukushima, and K. Murase, *Phys. Rev. D* **98**, 023019 (2018).
- [40] Y. Fujimoto, K. Fukushima, and K. Murase, *Phys. Rev. D* **101**, 054016 (2020).
- [41] S. Soma, L. Wang, S. Shi, H. Stöcker, and K. Zhou, *J. Cosmol. Astropart. Phys.* **08** (2022) 071.
- [42] F. Morawski and M. Bejger, *Astron. Astrophys.* **642**, A78 (2020).
- [43] P. G. Krastev, *Galaxies* **10**, 16 (2022).
- [44] P. G. Krastev, *Symmetry* **15**, 1123 (2023).
- [45] F. Morawski and M. Bejger, *Phys. Rev. C* **106**, 065802 (2022).
- [46] A. Thete, K. Banerjee, and T. Malik, *Phys. Rev. D* **108**, 063028 (2023).
- [47] G. Gonçalves, M. Ferreira, J. a. Aveiro, A. Onofre, F. F. Freitas, C. Providência, and J. A. Font, *J. Cosmol. Astropart. Phys.* **12** (2023) 001.
- [48] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, and M. Bennamoun, *IEEE Comput. Intell. Mag.* **17**, 29 (2022).
- [49] A. Olivier, M. D. Shields, and L. Graham-Brady, *Comput. Methods Appl. Mech. Eng.* **386**, 114079 (2021).
- [50] S. Typel and H. H. Wolter, *Nucl. Phys.* **A656**, 331 (1999).
- [51] M. Dutra, O. Lourenço, S. S. Avancini, B. V. Carlson, A. Delfino, D. P. Menezes, C. Providência, S. Typel, and J. R. Stone, *Phys. Rev. C* **90**, 055203 (2014).
- [52] S. Shlomo, V. M. Kolomietz, and G. Colò, *Eur. Phys. J. A* **30**, 23 (2006).
- [53] B. G. Todd-Rutel and J. Piekarewicz, *Phys. Rev. Lett.* **95**, 122501 (2005).
- [54] R. Essick, P. Landry, A. Schwenk, and I. Tews, *Phys. Rev. C* **104**, 065804 (2021).
- [55] I. Bombaci and U. Lombardo, *Phys. Rev. C* **44**, 1892 (1991).
- [56] I. Vidaña, C. m. c. Providência, A. Polls, and A. Rios, *Phys. Rev. C* **80**, 045806 (2009).
- [57] P. B. de Tovar, M. Ferreira, and C. Providência, *Phys. Rev. D* **104**, 123036 (2021).
- [58] D. P. Kingma and J. Ba, [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- [59] S. J. Reddi, S. Kale, and S. Kumar, [arXiv:1904.09237](https://arxiv.org/abs/1904.09237).

- [60] S. Bollweg, M. Haußmann, G. Kasieczka, M. Luchmann, T. Plehn, and J. Thompson, *SciPost Phys.* **8**, 006 (2020).
- [61] M. Abadi *et al.*, TensorFlow: Large-scale machine learning on heterogeneous systems, (2015), software available from tensorflow.org.
- [62] F. Chollet *et al.*, Keras, <https://keras.io> (2015).
- [63] M. B. Tsang *et al.*, *Phys. Rev. C* **86**, 015803 (2012).
- [64] J. M. Lattimer and Y. Lim, *Astrophys. J.* **771**, 51 (2013).
- [65] M. Oertel, M. Hempel, T. Klähn, and S. Typel, *Rev. Mod. Phys.* **89**, 015007 (2017).
- [66] M. Fortin, C. Providencia, A. R. Raduta, F. Gulminelli, J. L. Zdunik, P. Haensel, and M. Bejger, *Phys. Rev. C* **94**, 035804 (2016).
- [67] S. Soma, L. Wang, S. Shi, H. Stöcker, and K. Zhou, *Phys. Rev. D* **107**, 083028 (2023).
- [68] B.-A. Li and X. Han, *Phys. Lett. B* **727**, 276 (2013).
- [69] B.-A. Li, B.-J. Cai, W.-J. Xie, and N.-B. Zhang, *Universe* **7**, 182 (2021).
- [70] J. Margueron, R. Hoffmann Casali, and F. Gulminelli, *Phys. Rev. C* **97**, 025806 (2018).
- [71] C. Mondal, B. K. Agrawal, J. N. De, S. K. Samaddar, M. Centelles, and X. Viñas, *Phys. Rev. C* **96**, 021302 (2017).
- [72] M. Ferreira, M. Fortin, T. Malik, B. K. Agrawal, and C. Providência, *Phys. Rev. D* **101**, 043021 (2020).