Decoding neutron star observations: Revealing composition through Bayesian neural networks

Valéria Carvalho,[*] Márcio Ferreira,[†] Tuhin Malik,[‡] and Constança Providência[§]

CFisUC, Department of Physics, University of Coimbra, P-3004–516 Coimbra, Portugal

(Received 10 June 2023; accepted 3 August 2023; published 28 August 2023)

We exploit the great potential offered by Bayesian neural networks (BNNs) to directly decipher the internal composition of neutron stars (NSs) based on their macroscopic properties. By analyzing a set of simulated observations, namely NS radius and tidal deformability, we leverage BNNs as effective tools for inferring the proton fraction and sound speed within NS interiors. To achieve this, several BNNs models were developed upon a dataset of ~25 K nuclear equation of state within a relativistic mean-field framework, obtained through Bayesian inference that adheres to minimal low-density constraints. Unlike conventional neural networks, BNNs possess an exceptional quality: they provide a prediction uncertainty measure. To simulate the inherent imperfections present in real-world observations, we have generated four distinct training and testing datasets that replicate specific observational uncertainties. Our initial results demonstrate that BNNs successfully recover the composition with reasonable levels of uncertainty. Furthermore, using mock data prepared with the DD2, a different class of relativistic mean-field model utilized during training, the BNN model effectively retrieves the proton fraction and speed of sound for neutron star matter.

DOI: 10.1103/PhysRevD.108.043031

I. INTRODUCTION

The extreme matter conditions inside neutron stars (NSs) are impossible to recreate in terrestrial laboratories, making the equation of state (EOS) of dense and asymmetric nuclear matter (realized inside NSs) an interesting and still unknown quantity. Modeling NS matter is restricted by constraints coming from the observations of massive NSs: PSR J1614-2230 [1–3] with $M = 1.908 \pm 0.016 M_\odot$, PSR J0348–0432 with $M = 2.01 \pm 0.04 M_\odot$ [4], PSR J0740 + 6620 with $M = 2.08 \pm 0.07 M_\odot$ [5], and J1810 + 1714 with $M = 2.13 \pm 0.04 M_\odot$ [6]. Additionally, theoretical calculations, such as chiral effective field theory (χEFT), are applicable only at very low densities, while perturbative quantum chromodynamics is reliable for extremely high densities. Recently, multimessenger astrophysics has become an exciting field, allowing to understand NSs physics by connecting information carried by different sources, such as gravitational waves, photons, and neutrinos. The detection by LIGO/Virgo collaboration of compact binary coalescence events, such as GW170817 [7] and GW190425 [8], allowed us to further constrain the EOS of NS matter. Recent results from Neutron star Interior Composition ExploreR (NICER) on the PSR J0030 + 045 pulsar mass [9,10] and PSR J0740 + 6620 radius [11–13] were relevant in restricting the possible neutron stars physics. Future expected observations from experiments such as the enhanced x-ray timing and polarimetry mission (eXTP) [14,15], the (STROBE-X) [16], and Square Kilometer Array [17] telescope will allow for the determination of NSs radii and masses with a few % uncertainty.

Numerous statistical methods have been extensively explored to determine the most probable EOS based on observational data of NSs. These methods include Bayesian inference [18,19] and Gaussian processes [20]. However, even if the EOS is known with high precision, the challenge remains in constraining the composition of neutron star matter. Previous studies have highlighted the impossibility of recovering nuclear matter properties solely from the β-equilibrium EOS without knowledge of the compositions (or symmetry energy at high densities) [21–23] or without information about the EOS of symmetric nuclear matter in conjunction with compositions [24]. However, these studies were either limited to metamodels or based on restricted models. Motivated by this, we aim to construct an artificial neural network that directly maps NS observational properties to EOS composition using a large set of EOS derived from the relativistic mean field (RMF) approach.

Deep learning is another field that has become a buzzword in trying to solve the dense matter EOS problems, and all kinds of physics problems [25–44]. The inference problem of determining the EOS from

[*] val.mar.dinis@gmail.com

[†] marcio.ferreira@uc.pt

[‡] tm@uc.pt

[§] cp@uc.pt

observational data can be roughly divided into two main categories: the reconstructing of the EOS, pressure or speed of sound, from either mass radius or tidal deformability, [25–32,34,35,37,40,41,44], or focus directly on the nuclear matter saturation properties [33,36,38,39,42,43]. As an example, Fujimoto *et al.* [25–27] explored a framework based on neural networks (NNs) where an observational set of NS mass, radius, and respective variances was used as input and the speed of sound squared as output. A similar perspective was followed in [30]. The works [33,36] fall into the second category: determining specific saturation properties of nuclear matter, in this case, the density dependence of the nuclear symmetry energy, directly from observational NS data.

However, the majority of these NNs based models face a considerable drawback, namely, the lack of uncertainty quantification. Questions such as how confident is a model about its predictions? is the main focus of the present work, in which the uncertainty modeling is explored by implementing a very appealing approach called Bayesian neural networks (BNNs), [45]. BNNs have already started being used in different fields of physics [46,47], and have been shown useful in uncertainty quantification. Our goal is to implement an inference framework that gives a prediction uncertainty to any model prediction. We analyze the density dependence of the proton fraction and speed of sound inside NS matter. For that, several synthetic datasets made of "mock" observational data will be constructed and the impact of adding information on the tidal deformability into the model predictions will be analyzed. Let us make clear an important distinction between our work and the majority of studies: instead of applying a widely used approach of parametrizing the EOS, e.g., with polytropes, we used a specific family of nuclear models to construct a set of possible EOS. Despite their flexibility in exploring the entire region of possible EOS, the generic agnostic parametrizations of the EOS are unable to model and track the different degrees of freedom inside NSs. The use of a microscopic model has the crucial advantage of accessing the density dependence of all degrees of freedom and thus unable us to analyze the proton fraction.

The paper is organized as follows. A basic introduction to BNNs is presented in Sec. II. The family of nuclear models chosen is presented in Sec. III and also the Bayesian inference framework employed to construct the EOS dataset. The generation of the synthetic observation datasets is explained in Sec. IV. The model results for the proton fraction and speed of sound are discussed in Sec. V, and lastly, the conclusions are drawn in Sec. VII.

II. BAYESIAN NEURAL NETWORKS

Despite the great success of (feedforward) neural networks (NNs) in different fields of science, they come with some drawbacks that require special attention. NNs are susceptible to overfitting and are unable to access the uncertainty of its predictions, which may lead to overconfident predictions. Bayesian neural networks (BNNs) is a Bayesian approach framework that introduces stochastic weights to NNs making them uncertainty-aware models [48].

NNs are capable of representing arbitrary functions and are composed, in their simplest architecture, of a sequence of blocks where a linear transformation is followed by a nonlinear operation (activation functions). To simplify the notation, let us denote a NN by $\boldsymbol{y} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$, where $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{b})$ represent all NN weights. The vectors $\boldsymbol{W}$ and $\boldsymbol{b}$ denote, respectively, the connections (weights) and bias of all linear transformations of the network, which define completely the NN model. Training the NN consists in determining the numerical procedure (back-propagation algorithm) of finding the $\boldsymbol{\theta}^*$ that minimizes a chosen cost function on the training data. This traditional approach of estimating a single model defined by $\boldsymbol{\theta}^*$ ignores all other possible parametrizations $\boldsymbol{\theta}$.

BNNs simulate multiple possible NNs models by introducing stochastic weights. These networks operate by first choosing a functional model, i.e., a network architecture, and then the stochastic model, i.e., the probability distributions for the weights. Bayesian inference is then required to train the network by defining the likelihood function of the observed data, $P(D|\boldsymbol{\theta})$, and the prior probability distribution over the model parameters, $P(\boldsymbol{\theta})$. It is then possible to employ Bayes theorem and obtain the posterior probability distribution, i.e., the probability of the model parameters given the data:

$$P(\boldsymbol{\theta}|D) = \frac{P(D|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(D)}, \tag{1}$$

where $P(D) = \int_{\boldsymbol{\theta}'} P(D|\boldsymbol{\theta}')P(\boldsymbol{\theta}')d\boldsymbol{\theta}'$ is the evidence. Having a distribution on the weights, the BNNs predictions become a Bayesian model average: the probability distribution of some unknown $\boldsymbol{y}^*$ given an input $\boldsymbol{x}^*$ is

$$P(\boldsymbol{y}^*|\boldsymbol{x}^*, D) = \int_{\boldsymbol{\theta}} P(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta})P(\boldsymbol{\theta}|D)d\boldsymbol{\theta}. \tag{2}$$

$P(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta})$ is considered to be the likelihood of our data, the distribution that comes out of the network and captures the noise present in our data, and $P(\boldsymbol{\theta}|D)$ is the posterior distribution of our weights, that brings up the uncertainty on the model. Another advantage of using these networks is that they capture two types of uncertainty, aleatoric uncertainty, uncertainty on the data, and epistemic uncertainty, uncertainty on the model estimation defined as $P(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta})$ and $P(\boldsymbol{\theta}|D)$, respectively. However, solving Eq. (2) is a very complex task because the posterior $P(\boldsymbol{\theta}|D)$ depends on the evidence $P(D) = \int_{\boldsymbol{\theta}'} P(D|\boldsymbol{\theta}')P(\boldsymbol{\theta}')d\boldsymbol{\theta}'$, which is a nonanalytic expression that requires marginalizing over all model parameters. Fortunately, there are multiple ways of tracking it by using either Markov chain Monte Carlo or variational inference (more information can be found in [48]). We are going to implement the variational inference method that is presented in the following.

A. Varitional inference formalism

The variational inference method aims to approximate a variational posterior $q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ to the real posterior $P(\boldsymbol{\theta}|D)$ by using the Kullback-Leibler (KL) divergence. The KL divergence is a measure of dissimilarity between two probability distributions. It approaches zero when the variational posterior and the true posterior are identical and is positive otherwise. Fundamentally, we want to find the variational posterior that corresponds to the minimum value of the KL divergence between the variational posterior and the true posterior:

$$q_{\phi^*} = \underset{q_\phi}{\mathrm{argmin}}\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\theta})||P(\boldsymbol{\theta}|D)), \tag{3}$$

where the KL divergence is defined by

$$\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\theta})||P(\boldsymbol{\theta}|D)) = \mathbb{E}_{q_\phi(\boldsymbol{\theta})}\left[\log\left(\frac{q_\phi(\boldsymbol{\theta})}{P(\boldsymbol{\theta}|D)}\right)\right], \tag{4}$$

$$= \int_{\boldsymbol{\theta}} q_\phi(\boldsymbol{\theta}) \log\left(\frac{q_\phi(\boldsymbol{\theta})}{P(\boldsymbol{\theta}|D)}\right) d\boldsymbol{\theta}. \tag{5}$$

In order for the dependence on the true posterior to disappear, the last equation can be rewritten, with the help of Bayes rule Eq. (1), as

$$\begin{aligned}\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\theta})||P(\boldsymbol{\theta}|D)) &= \int_{\boldsymbol{\theta}} q_\phi(\boldsymbol{\theta}) \log\left(\frac{q_\phi(\boldsymbol{\theta})P(D)}{P(D|\boldsymbol{\theta})P(\boldsymbol{\theta})}\right) d\boldsymbol{\theta},\\ &= \mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\theta})||P(\boldsymbol{\theta}))\\ &\quad - \mathbb{E}_{q_\phi(\boldsymbol{\theta})}(\log P(D|\boldsymbol{\theta})) + \log P(D),\\ &= F(D,\boldsymbol{\phi}) + \log P(D),\end{aligned} \tag{6}$$

where $F(D,\boldsymbol{\phi}) = \mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\theta})||P(\boldsymbol{\theta})) - \mathbb{E}_{q_\phi(\boldsymbol{\theta})}(\log P(D|\boldsymbol{\theta}))$ is called the variational free energy. We end up with

$$\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\theta})||P(\boldsymbol{\theta}|D)) = F(D,\boldsymbol{\phi}) + \log P(D). \tag{7}$$

As the last term, $\log P(D)$, does not depend on the variational posterior and his parameters, which we are optimizing, minimizing $\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\theta})||P(\boldsymbol{\theta}|D))$ with respect to $\boldsymbol{\phi}$ is the same as minimizing $F(D,\boldsymbol{\phi})$. Evidence Lower Bound (ELBO) is another important quantity, which stands for evidence lower bound, and it is defined as the negative free energy, i.e., $\mathrm{ELBO} = -F(D,\boldsymbol{\phi})$. Equation (7) can then be rewritten as

$$\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\theta})||P(\boldsymbol{\theta}|D)) = -\mathrm{ELBO} + \log P(D), \tag{8}$$

or

$$\mathrm{ELBO} = -\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\theta})||P(\boldsymbol{\theta}|D)) + \log P(D). \tag{9}$$

ELBO is called the lower bound of the evidence because $\mathrm{ELBO} \leq \log P(D)$. In other words, $\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\theta})||P(\boldsymbol{\theta}|D))$ is minimized by maximizing the evidence lower bound.

In the end, our optimization objective resumes to

$$\begin{aligned}q_{\phi^*} &= \underset{q_\phi}{\arg\min}\ \mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\theta})||P(\boldsymbol{\theta}|D)),\\ &= \underset{q_\phi}{\arg\max}\ \mathrm{ELBO} = \underset{q_\phi}{\arg\min}\ F(D,\theta),\\ &= \underset{q_\phi}{\arg\min}\ [\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\theta})||P(\boldsymbol{\theta})) - \mathbb{E}_{q_\phi(\boldsymbol{\theta})}(\log P(D|\boldsymbol{\theta}))].\end{aligned}$$

The above general formalism is applied to our specific case, where we have chosen a multivariate Gaussian for the variational posterior, $q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$, and a multivariate Gaussian with diagonal covariance matrix for the prior, $P(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$.[1] The final loss function we are trying to minimize uses Monte Carlo sampling to obtain the expected values, where $\boldsymbol{\theta}^{(n)}$ is being sampled from the variational posterior, $q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$, and for our specific model, also the exact value of the KL divergence between two Gaussian's with covariance matrix is used,

$$\begin{aligned}F(D,\boldsymbol{\phi}) = &\frac{1}{2D_s}[(-\log\det(\boldsymbol{\Sigma}_q)) - k + \mathrm{tr}(\boldsymbol{\Sigma}_q) + (\boldsymbol{\mu}_q)^T(\boldsymbol{\mu}_q)]\\ &- \frac{1}{B}\sum_{i=1}^{B}\frac{1}{N}\sum_{n=1}^{N}\log P(\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{\theta}^{(n)}),\end{aligned} \tag{10}$$

where B is the number of points of the mini-batch, D_s is the number 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 of the variational posterior (we used $N = 10^4$). One aspect that stands out in these networks is how back-propagation works. Without going into much detail, the back-propagation updates $\boldsymbol{\mu}_q$ and $\boldsymbol{\Sigma}_q$ for each of the networks' parameters (more information can be found in [49]). Once the network is trained and the best mean $\boldsymbol{\mu}_q$ and covariance matrix $\boldsymbol{\Sigma}_q$ are obtained for the distribution of the parameters, Eq. (2) becomes solvable and predictions are obtained using Monte Carlo estimations

$$P(\boldsymbol{y}^*|\boldsymbol{x}^*, D) = \int_{\boldsymbol{\theta}} P(\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta}) q_\phi(\boldsymbol{\theta}) d\boldsymbol{\theta}, \tag{11}$$

$$= \frac{1}{N}\sum_{n=1}^{N} P(\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta}^{(n)}), \quad \boldsymbol{\theta}^{(n)} \sim q_\phi(\boldsymbol{\theta}). \tag{12}$$

[1] The choice of different probabilistic distributions and its impact on the inference results is left for future studies.

The mean $\hat{\boldsymbol{\mu}}$ and variance $\hat{\boldsymbol{\sigma}}^2$ vectors of the predicting distribution $P(\boldsymbol{y}^*|\boldsymbol{x}^*, D)$ can be calculated, for a fixed $\boldsymbol{x}^*$, by applying the law of total expectation and total variance. From

$$\mathbb{E}[\boldsymbol{y}^*|\boldsymbol{x}^*, D] = \mathbb{E}_{q_\phi(\boldsymbol{\theta})}[\mathbb{E}[\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta}]] \tag{13}$$

and

$$\mathrm{Var}[\boldsymbol{y}^*|\boldsymbol{x}^*, D] = \mathbb{E}_{q_\phi(\boldsymbol{\theta})}[\mathrm{Var}[\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta}]] - \mathrm{Var}_{q_\phi(\boldsymbol{\theta})}[\mathbb{E}[\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta}]], \tag{14}$$

we obtain

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{n=1}^{N} \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}_n} \tag{15}$$

and

$$\hat{\boldsymbol{\sigma}}^2 = \underbrace{\frac{1}{N}\sum_{n=1}^{N} \hat{\boldsymbol{\sigma}}^2_{\boldsymbol{\theta}_n}}_{\text{Aleatoric uncertainty}} + \underbrace{\frac{1}{N}\sum_{n=1}^{N} (\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}_n} - \hat{\boldsymbol{\mu}}) \odot (\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}_n} - \hat{\boldsymbol{\mu}})}_{\text{Epistemic uncertainty}}, \tag{16}$$

where $\odot$ denotes element-wise multiplication. The predicted variance captures both epistemic and aleatoric uncertainties [50].

III. NUCLEAR MODELS

A field theoretical approach is adopted to calculate a dataset of nuclear EOSs. The approach incorporates self-interactions and mixed meson terms within a RMF description. A wide and reasonable region of the parameter space is considered, providing an accurate representation of presently known nuclear properties. Including nonlinear terms is crucial for determining the density dependence of the EOS. In this treatment, the nucleons interact through the exchange of scalar-isoscalar mesons (σ), vector-isoscalar mesons (ω), and vector-isovector mesons (ρ). The Lagrangian governing the baryonic degrees of freedom can be expressed as follows: $\mathcal{L} = \mathcal{L}_N + \mathcal{L}_M + \mathcal{L}_{NL}$ with

$$\begin{aligned}
\mathcal{L}_N &= \bar{\Psi}[\gamma^{\boldsymbol{\mu}}(i\partial_{\boldsymbol{\mu}} - g_\omega \omega_{\boldsymbol{\mu}} - g_\varrho \mathbf{t}\cdot \varrho_{\boldsymbol{\mu}}) - (m - g_\sigma \sigma)]\Psi, \\
\mathcal{L}_M &= \frac{1}{2}[\partial_{\boldsymbol{\mu}}\sigma\partial^{\boldsymbol{\mu}}\sigma - m_\sigma^2\sigma^2] - \frac{1}{4}F^{(\omega)}_{\boldsymbol{\mu\nu}}F^{(\omega)\boldsymbol{\mu\nu}} + \frac{1}{2}m_\omega^2\omega_{\boldsymbol{\mu}}\omega^{\boldsymbol{\mu}} \\
&\quad - \frac{1}{4}\boldsymbol{F}^{(\varrho)}_{\boldsymbol{\mu\nu}}\cdot \boldsymbol{F}^{(\varrho)\boldsymbol{\mu\nu}} + \frac{1}{2}m_\varrho^2 \varrho_{\boldsymbol{\mu}}\cdot \varrho^{\boldsymbol{\mu}}, \\
\mathcal{L}_{NL} &= -\frac{1}{3}bmg_\sigma^3(\sigma)^3 - \frac{1}{4}cg_\sigma^4(\sigma)^4 + \frac{\xi}{4!}g_\omega^4(\omega_{\boldsymbol{\mu}}\omega^{\boldsymbol{\mu}})^2 \\
&\quad + \Lambda_\omega g_\varrho^2 \varrho_{\boldsymbol{\mu}}\cdot \varrho^{\boldsymbol{\mu}} g_\omega^2 \omega_{\boldsymbol{\mu}}\omega^{\boldsymbol{\mu}},
\end{aligned}$$

where the field Ψ denotes the Dirac spinor that describes the nucleon doublet (neutron and proton) with a bare mass m, $\gamma^{\boldsymbol{\mu}}$ are the Dirac matrices, and $\boldsymbol{t}$ is the isospin operator. The vector meson tensors are defined as $F^{(\omega,\varrho)\boldsymbol{\mu\nu}} = \partial^{\boldsymbol{\mu}}A^{(\omega,\varrho)\nu} - \partial^\nu A^{(\omega,\varrho)\boldsymbol{\mu}}$. The couplings of the nucleons to the meson fields σ, ω, and ϱ are denoted by g_σ, g_ω, and g_ϱ, respectively. The meson masses are given by m_σ, m_ω, and m_ϱ. More information on the specifics of the model can be found in [19] and references therein. The parameters g_σ, g_ω, g_ρ, b, c, ξ, and Λ_ω are systematically sampled within a Bayesian framework, adhering to minimal constraints imposed by several nuclear saturation properties. Furthermore, these parameters are subject to the conditions of the neutron star maximum mass exceeding $2M_\odot$, as well as the EOS for low-density pure neutron matter, which is meticulously generated through a precise N^{3}LO calculation in chiral effective field theory. A detailed discussion on these aspects will be presented in the subsequent subsection.

A. The Bayesian setup

Based on observed or fitted data, a prior belief (expressed as a prior distribution) is updated using Bayesian inference. The posterior distribution is derived according to Bayes' theorem [51]. In order to establish a Bayesian parameter optimization system, four key components must be defined: the prior, the likelihood function, the fit data, and the sampler.

1. The prior

A broad range of nuclear matter saturation properties is carefully considered in the prior domain of the adopted RMF model. As a result of Latin hypercube sampling, we determine the prior range in our Bayesian setup. Uniform priors are chosen for each parameter, as described in Table I.

2. The fit data

The fit data, presented in Table II, include the nuclear saturation density ρ_0, the binding energy per nucleon ϵ_0, the

TABLE I. We use a uniform prior range for the parameters of the RMF models. Specifically, B and C are $b \times 10^3$ and $c \times 10^3$, respectively. Distribution minimums and maximums are indicated by “min” and “max,” respectively.

No	Parameters	*Set 0*	
		Min	Max
1	g_σ	6.5	15.5
2	g_ω	6.5	15.5
3	g_ϱ	6.5	16.5
4	B	0.5	9.0
5	C	−5.0	5.0
6	ξ	0.0	0.04
7	Λ_ω	0	0.12

TABLE II. The Bayesian inference imposes constraints on various quantities to generate sets of models. These constraints include the binding energy per nucleon ϵ_0, incompressibility K_0, and symmetry energy $J_{\rm sym,0}$ at the nuclear saturation density ρ_0, each with a 1σ uncertainty. Additionally, the pressure of PNM is considered at densities of 0.08, 0.12, and 0.16 fm^{-3}, obtained from a χEFT calculation [52]. The likelihood incorporates a 2 × N^{3}LO uncertainty for the PNM pressure, noting that it increases with density. Furthermore, the maximum mass of neutron stars is constrained to be above $2M_\odot$.

Constraints			
Quantity		Value/Band	References
NMP [MeV]	ρ_0	0.153 ± 0.005	[53]
	ϵ_0	-16.1 ± 0.2	[54]
	K_0	230 ± 40	[55,56]
	$J_{\rm sym,0}$	32.5 ± 1.8	[57]
PNM [MeV fm^{-3}]	$P(\rho)$	$2 \times$ N^{3}LO	[52]
	$dP/d\rho$	> 0	
NS mass [$M_\odot$]	$M_{\rm max}$	> 2.0	[5]

incompressibility coefficient K_0, and the symmetry energy $J_{\rm sym,0}$, all evaluated at ρ_0. Additionally, we incorporate the pressure of pure neutron matter (PNM) at densities of 0.08, 0.12, and 0.16 fm^{-3} from N^{3}LO calculations in χEFT [52], accounting for 2 × N^{3}LO data uncertainty. Furthermore, the likelihood also includes the requirement of the neutron star maximum mass exceeding $2.0M_\odot$ with uniform probability.

3. The log-likelihood

We optimize a log-likelihood function as a cost function for the given fit data in Table II. Equation (17) represents the log-likelihood function, taking into account the uncertainties σ_j associated with each data point j of constraints. The maximum mass of neutron stars is treated differently, using step function probability,

$$\mathrm{Log}(\mathcal{L}) \propto -\sum_j \left\{ \left(\frac{d_j - m_j(\boldsymbol{\theta})}{\sigma_j} \right)^2 + \mathrm{Log}(2\pi\sigma_j^2) \right\}. \tag{17}$$

To populate the seven-dimensional posterior, we employ the nested sampling algorithm [58], specifically the PyMultinest sampler [59,60], which is well suited for low-dimensional problems. The EOS dataset for subsequent analyses will be generated using the full posterior, which contains 25287 EOS.

IV. DATASET

Our goal is to train BNNs models to predict the speed of sound and proton fraction of NS matter from a given set of NS mock observations. To understand the effect of different NS properties on the prediction uncertainty, we generate different datasets using the 25287 EOS that were obtained through the Bayesian analysis formalism (see Sec. III A).

A. Structure

Our BNN model, $P(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta})$, attributes a probability distribution to $\boldsymbol{Y}$ (output space) given a set of NS mock observations $\boldsymbol{X}$ (input space), where $\boldsymbol{Y}$ denotes the different NS matter properties under study, i.e., the speed of sound $\boldsymbol{v_s^2(n)}$ and proton fraction $\boldsymbol{y_p(n)}$. We have chosen to characterize each element of $\boldsymbol{Y}$ at 15 fixed baryonic densities n_k, e.g., $\boldsymbol{y_p(n)} = [y_p(n_1), y_p(n_2), \ldots, y_p(n_{15})]$. The density points are equally spaced between $n_1 = 0.15$ fm^{-3} and $n_{15} = 1.0$ fm^{-3}, $n_k = \{0.15, 0.21, 0.27, \ldots, 1.0\}$ fm^{-3}. The number of points ($N_Y = 15$) was selected as a trade-off between computational training time and the interpolation accuracy, i.e., low residuals between the interpolation and the real values. We illustrate this discretization process of the output space elements for the proton fraction in Fig. 1.

Regarding the structure of the input space $\boldsymbol{X}$, two different element structures are studied: (i) $\boldsymbol{X} = [M_1, \ldots, M_5, R_1, \ldots, R_5]$ corresponding to five $M_i(R_i)$ simulated observations and (ii) $\boldsymbol{X} = [M_1, \ldots, M_5, R_1, \ldots, R_5, M'_1, \ldots, M'_5, \Lambda_1, \ldots, \Lambda_5]$ corresponding to five $M_i(R_i)$ and five $\Lambda_j(M'_j)$ simulated observations. In summary, the output elements $\boldsymbol{Y}_i$ of our datasets are specified by 15-dimensional vectors and the input space elements $\boldsymbol{X}_i$ by 10-dimensional or 20-dimensional vectors, depending on the dataset type under study. The statistical procedure for generating the different synthetic observational datasets is presented in the following.

B. Generation

The first step of the generation of the datasets consists in randomly splitting the total number of EOS into train and test sets in a proportion of 80%/20%, i.e., the train set contains 22758 EOS while the test has 2529 EOS. Second,

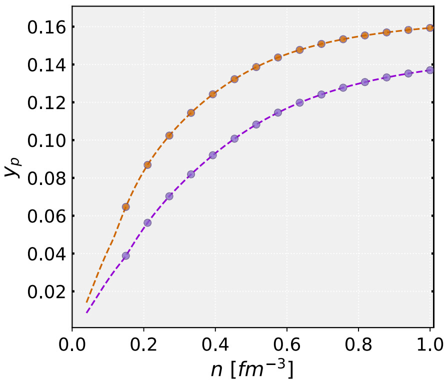

FIG. 1. Illustration of $\boldsymbol{Y} = \boldsymbol{y_p(n)}$ for two EOSs: each EOS is represented in our datasets by the 15 points $y_p(n_k)$.

we generate two types of datasets that share the $\boldsymbol{Y}_i$ structure but with different $\boldsymbol{X}_i$ structures:

$$\boldsymbol{X_i} = [M_1, \ldots, M_5, R_1, \ldots, R_5],$$
$$\boldsymbol{X'_i} = [M_1, \ldots, M_5, R_1, \ldots, R_5, M'_1, \ldots, M'_5, \Lambda_1, \ldots, \Lambda_5].$$

These different structures allows us to compare and access how informative is the tidal deformability on the model predictions. The statistical generating procedure is composed of the following steps. For each EOS, we randomly select five NS mass values, $M_i^{(0)}$, from a uniform distribution between $1.0M_\odot$ and $M_{\max}$. Then, the radius R_i is sampled from a Gaussian distribution centred at the TOV solution, denoted as $R(M_i^{(0)})$, and with a standard deviation of σ_R. Finally, we sample the final NS mass from a Gaussian distribution centered at $M_i^{(0)}$ and a standard deviation of σ_M. The above process can be summarized in the following equations:

$$M_i^{(0)} \sim \mathcal{U}[1, M_{\max}] \quad (\text{in units of } M_\odot), \tag{18}$$

$$R_i \sim \mathcal{N}(R(M_i^{(0)}), \sigma_R^2), \tag{19}$$

$$M_i \sim \mathcal{N}(M_i^{(0)}, \sigma_M^2), \quad i = 1, \ldots, 5. \tag{20}$$

The final generated elements consists of $\boldsymbol{X_i} = [M_1, \ldots, M_5, R_1, \ldots, R_5]$, and each one is a possible realization (*observation*) that characterizes the $M(R)$ diagram of the specific EOS. This procedure is similar to the one used in [25], where a Gaussian noise was applied to 15 values from $M(R)$ curve, and then shifted from the original mass radius curve: $M_i = M_i^{(0)} + \mathcal{N}(0, \sigma_M^2)$ and $R_i = R(M_i^{(0)}) + \mathcal{N}(0, \sigma_R^2)$, for $i = 1, \ldots, 15$. The second kind of datasets includes the tidal deformability and has the additional steps

$$M'_j \sim \mathcal{U}[1, M_{\max}] \quad (\text{in units of } M_\odot), \tag{21}$$

$$\Lambda_j \sim \mathcal{N}(\Lambda(M'_j), \sigma_\Lambda^2(M'_j)) \quad j = 1, \ldots, 5, \tag{22}$$

where $\Lambda(M'_j)$ is given by the $\Lambda(M)$ relation of the specific EOS and $\sigma_\Lambda(M'_j)$ describes an overall dispersion around the mean value. Samples values with $\Lambda < 0$ were discarded. The functional form $\sigma_\Lambda(M'_j)$ should reflect our expectation on the mock observational uncertainty as a function of the NS mass. In the present work, we considered $\sigma_\Lambda(M'_j) = \text{constant} \times \hat{\sigma}(M'_j)$, where $\hat{\sigma}(M'_j)$ is the standard deviation of $\Lambda(M)$ determined from dataset of EOSs. The generated point is $\boldsymbol{X_i} = [M_1, \ldots, M_5, R_1, \ldots, R_5, M'_1, \ldots, M'_5, \Lambda_1, \ldots, \Lambda_5]$ (a similar approach can be found in [37]). In the above procedures, there is an additional parameter, which we denote by n_s, that specifies the number of mock observations for each EOS, i.e., the number of times the above procedures are applied to each EOS. For instance, choosing $n_s = 20$ would mean running the above procedures 20 times for each EOS (20 *observations*), and thus obtaining $\{\boldsymbol{X_1}, \boldsymbol{X_2}, \ldots, \boldsymbol{X_{20}}\}$.

TABLE III. Generation parameters for each dataset. $\hat{\sigma}(M_j)$ denotes the standard deviation of $\Lambda(M)$ calculated on the train set.

Dataset	σ_M $[M_\odot]$	σ_R [km]	$\sigma_\Lambda(M_j)$
1	0.05	0.15	$\cdots$
2	0.1	0.3	$\cdots$
3	0.1	0.3	$0.5\hat{\sigma}(M_j)$
4	0.1	0.3	$2\hat{\sigma}(M_j)$

Applying the above formalism to both the train and test sets, we have generated a total of four datasets whose properties are displayed in Table III. Sets 1 and 2 only contain information about the NS radii (input space $\boldsymbol{X}$ is 10-dimensional) while sets 3 and 4 also include the tidal deformability (input space $\boldsymbol{X}$ is 20-dimensional). The analysis of sets 1 and 2 allows to understand how a decrease in the spread of the mock observations around the TOV solution affects the predictions and uncertainties. In the same manner, sets 3 and 4 aim to understand possible effects on the model predictions arising from an increase of simulated observations scattering around the mean value on the tidal deformability. We use 60 mock observations, $n_s = 60$, on the training sets for each EOS while $n_s = 1$ was employed for the test sets. This key difference tries to simulate a real case scenario in which we only have access to a single mock observation of the *true* EOS. Here, by single mock observation, we mean $n_s = 1$ that corresponds to five $M_i(R_i)$ mock observations (sets 1 and 2) or five $M_i(R_i)$ mock observations and five $\Lambda_j(M_j)$ mock observations (sets 3 and 4). To illustrate the dataset generation, Fig. 2 displays the 60 mock observations for two distinct

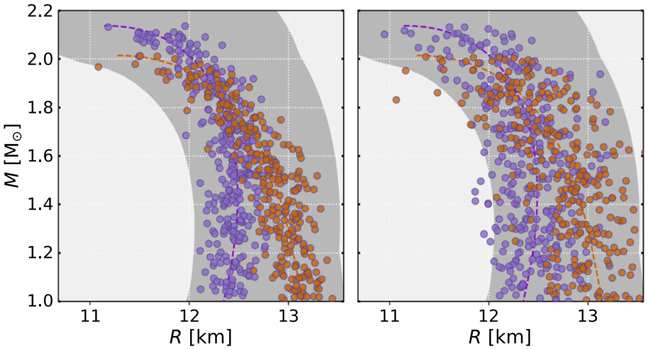

FIG. 2. The $n_s = 60$ mock observations generated for two EOSs in dataset 1 (left) and dataset 2 (right). The gray area represents the extremes of our EOS dataset. The two EOSs coincide with the ones used in Fig. 1.

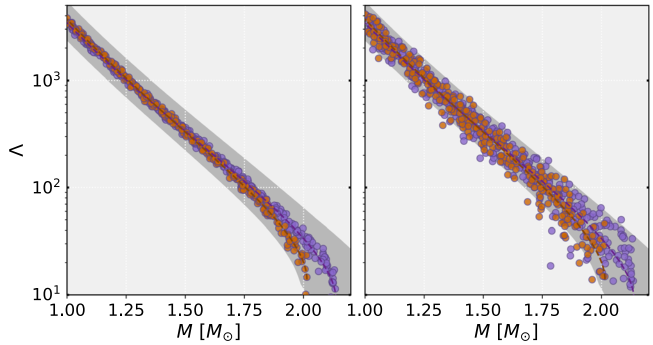

FIG. 3. The $n_s = 60$ mock observations generated in the $\Lambda - M$ diagram for two EOSs in dataset 3 (left) and dataset 4 (right). The gray area represents the extremes of our EOS dataset. The two EOSs coincide with the ones used in Fig. 1.

EOSs that belong to the generated datasets (dataset 1 and 2 on the right and left figures, respectively). Note that, for each EOS, there are 300 points in the $M(R)$ diagram, i.e., consisting of 60 EOS simulated observations for five NS mock observations each. It is clear the increase of σ_R from dataset 1 to 2, highlighting the differences between the two datasets. Furthermore, we provide Fig. 3 to depict the tidal deformability values for datasets 3 and 4.

C. Training procedure

To assess the response of BNNs to varying input noises and output targets, we have conducted experiments involving the training of diverse functional and stochastic models (as explained in Sec. II). These BNN models were trained using distinct datasets generated as outlined in Sec. IV B. During the training stage, a subset of the training data was randomly selected as validation step, i.e., the training data was split into 80% actually for training and 20% for validation. Moreover, the input data $\boldsymbol{X}$ was standardized.

Defining the functional models involves adjusting the number of neurons, layers, and activation functions. Table IV shows the best functional models for each dataset mentioned in Table III. For the hidden layers, we explore hyperbolic tangent, softplus, and sigmoid activation functions, while utilizing a linear activation function for the output layer. As the input vector sizes differ (10 for sets 1 and 2, and 20 for sets 3 and 4), we employ more neurons per layer for larger input spaces. This is due to the increased complexity demanded by a greater number of parameters. Specifically, we use 15 and 10 neurons for sets 1 and 2, and 20 and 25 neurons for sets 3 and 4, respectively. The output layer consistently contains 30 neurons, with 15 representing the mean and 15 representing the standard deviation of the output probability distribution function. It is worth noting that we deliberately excluded the use of correlation in the output layer due to the inferior performance observed when attempting to incorporate it. As a result, the output layer is solely focused on capturing the mean and standard deviation information of the output distribution. The architecture employed in this study involved utilizing two to three hidden layers. The number of neurons within each hidden layer remained consistent, but it varied depending on the size of the input vector, as explained earlier. During the hyper-parameter search process, we systematically explored four different architectures for each output. However, we narrowed our focus to datasets 1 and 3, as we specifically aimed to identify the most suitable architecture for the two different input sizes. The best outcomes are obtained by employing sigmoid as the activation function in the hidden layers, ensuring minimal loss and preventing divergence. Across all eight dataset configurations, i.e., two outputs (v_s^2 and y_p) and four datasets, see Table III, it was found that the optimal number of hidden layers is two. For sets 1 and 2, the best performance was achieved with 15 neurons in each hidden layer, while for sets 3 and 4, 25 neurons were utilized in each hidden layer. Detailed information on these configurations can be found in Table IV for the two output variables. During training, we employ a learning rate of 0.001 and utilize the ADAM optimizer [61] with the AMSgrad improvement [62]. The models are trained for 4000 epochs, with a minibatch size of 768.

TABLE IV. Structures of the final BNN models. The $v_s^2(n)$ and $y_p(n)$ models have the same structure.

Layers	Activation	Neurons	
		Datasets 1 & 2	Datasets 3 & 4
Input	N/A	10	20
Hidden Layer 1	Sigmoid	15	25
Hidden Layer 2	Sigmoid	15	25
Output	Linear	30	30

Regarding the stochastic model, we adopt a Gaussian prior with mean zero and standard deviation of one as mentioned in Sec. II A. 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. [46]. Additionally, we select a multivariate normal distribution as the variational posterior as explained in Sec. II A, initialized with mean 0 and a diagonal covariance matrix, where the standard deviation is equal to $\log(1 + \exp 0) = 0.693$. Furthermore, we opt for a deterministic output layer instead of a probabilistic one, as it has demonstrated improved results in our experiments. This decision is motivated by the fact that the deterministic output layer aligns better with the specific requirements of our model architecture and the nature of the problem we are addressing. All BNNs models were coded using TensorFlow library [63], more specifically we use Keras [64], an high-level API of the TensorFlow.

V. RESULTS: NEUTRON STAR MATTER PROPERTIES

In the following, we discuss the results for the speed of sound squared $v_s^2(n)$ and proton fraction $y_p(n)$. To analyze how the observational uncertainty in R and Λ affects the model confidence, we are going to compare the results from the different models of Table IV, which were trained on the different datasets presented in Table III. For the sake of simplicity, whenever we want to distinguish the different eight BNN models, we will, hereafter, refer to their training (data)sets, 1 to 4, and predicting quantity, $v_s^2(n)$ or $y_p(n)$. For instance, when we say the results of dataset 2 for $y_p(n)$, we mean the BNN model trained on dataset 2 that has the proton fraction as the target quantity.

A. Speed of sound

Let us start by illustrating the different BNNs predictions for the speed of sound squared on a randomly selected EOS from the test set. The results are displayed in Fig. 4, where the top panel shows sets 1 (blue) and 2 (orange) models while the bottom figure shows sets 3 (purple) and 4 (green). Model predictions are computed using Eq. (2), from which we show the mean values (solid lines) and 2σ regions (color regions). The upper plot clearly shows that the prediction uncertainty, characterized by the distribution's standard deviation σ, is smaller on set 1 than on set 2, and most importantly, the predicted mean values are close to the real values. The same pattern is seen in the lower figure: set 3 model (purple) has a lower prediction variance than set 4. While a deeper understanding of the overall behavior requires that we analyze the whole test set, Fig. 4 already indicates that the BNNs models are able to capture the characteristics of the different datasets (see Table III): the increased dispersion of the NS mock observations around the true values translate into a larger uncertainty when inferring the corresponding EOS properties.

To investigate how the models predictions behave over the entire test set, we define the normalized residuals' predictions as $\Gamma(n_k) = (v_s^2(n_k) - v_s^2(n_k)^{\mathrm{true}})/\sigma(n_k)$ and the dispersion by $\Sigma(n_k) = \sigma(n_k)$ at each of the prediction densities, i.e., $k = 1, \ldots, 15$. A summary statistics of both quantities over all EOS of the test set is shown in Fig. 5. The distribution of $\Gamma(n)$ (top panel), for all four datasets, has 50% of the values near zero, indicating that the median of the prediction values is unbiased. Furthermore, at the 2.3% and 97.7% of the cumulative percentage, we can see that the distribution lies between -2σ and 2σ, respectively, indicating that the prediction mean deviates from the true value less than 2σ 95.4% of the times. The fact that the distribution properties of $\Gamma(n)$ are similar across all datasets and independent of the density reveals that the BNNs models are correctly modeling the dispersion of the predictions considering the corresponding mean residuals at each density value. The value of $\Sigma(n)$ (bottom panel) grows with increasing density reflecting the training set statistics: the EOS dataset was generated by Bayesian inference where saturation properties were imposed, leading to a wider uncertainty at higher densities while low density regions are strongly constrained. From the $\Sigma(n)$ plot, we see that the whole distribution of the BNN model trained on set 1 (blue line) shifts to lower values, specifically on 2.3% and 50% of the values, showing that there is a considerable decrease in uncertainty when the dispersion of NS mock observations is reduced by a factor of 2, from ($\sigma_M = 0.1 M_\odot, \sigma_R = 0.3$ km) to ($\sigma_M = 0.05 M_\odot, \sigma_R = 0.15$ km) (see Table III).

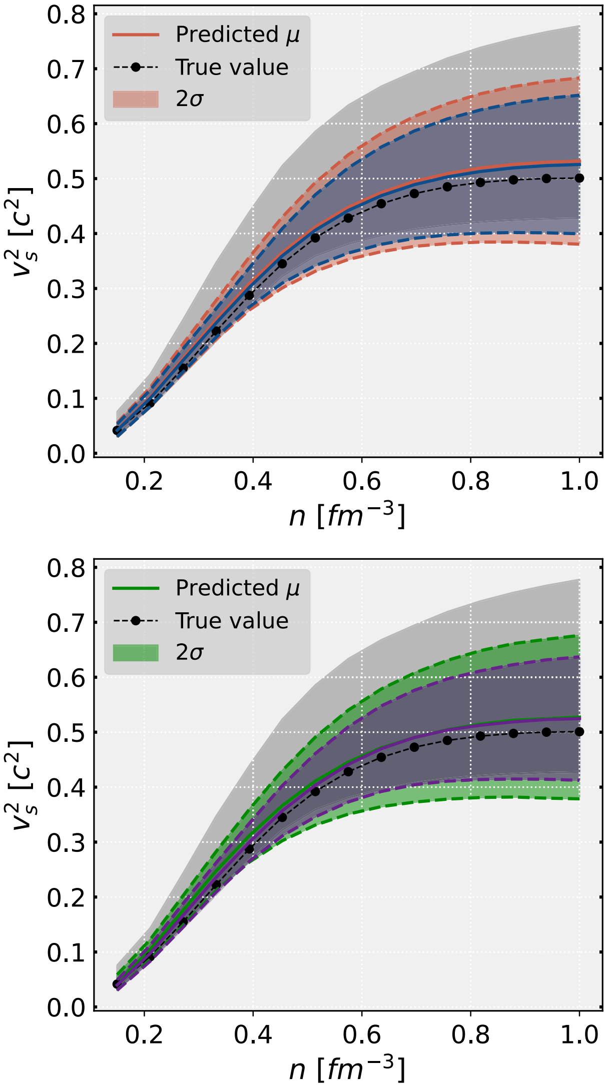

FIG. 4. The BNNs predictions for $v_s^2(n)$ using one EOS of the test. The models trained on datasets 1 (blue) and 2 (orange) are in the upper figure while datasets 3 (purple) and 4 (green) models are in the lower figure. The prediction mean values (solid lines) and 2σ confidence intervals are shown. The true values are shown in black dots and the range of $v_s^2(n)$ from the train set is indicated by the gray region.

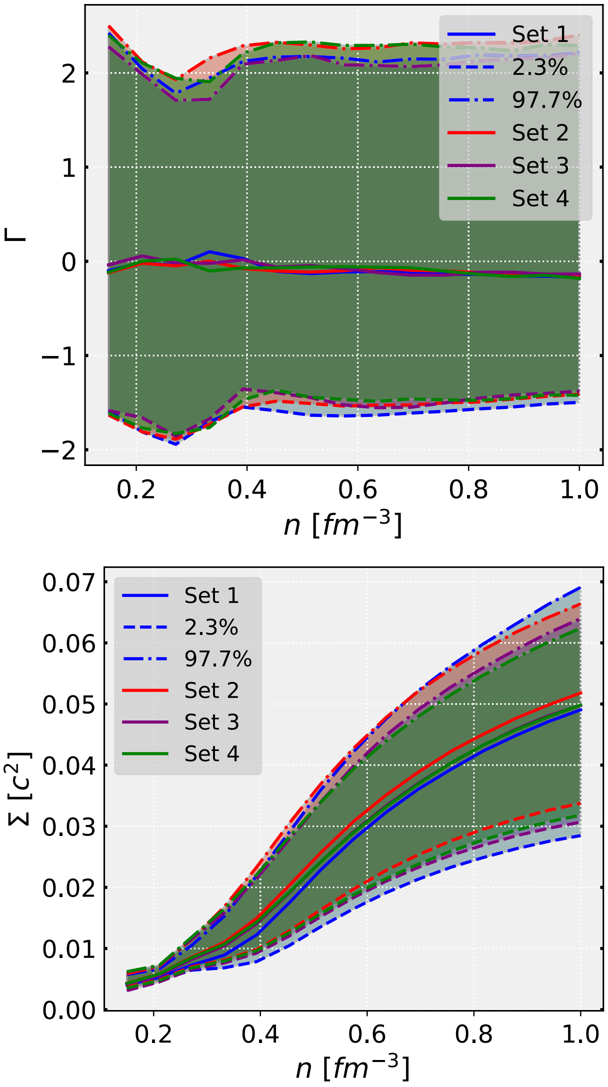

FIG. 5. Median (solid line), 95.4% confidence interval (dashed and dotted lines), and extreme values (region) for $\Gamma(n_k) = (v_s^2(n_k) - v_s^2(n_k)^{\mathrm{true}})/\sigma(n_k)$ (top) and $\Sigma(n_k) = \sigma(n_k)$ (bottom) for each dataset.

To estimate the overall performance of the different BNNs models, we show the coverage probability of each model in Fig. 6. The coverage probability quantifies how the model is perceiving the distribution of the data, by determining if the percentage of values contained in 1σ of the output distribution, i.e., the number of values in a specific interval divided by the total number of values, corresponds to 68% of the number of values we are using for the test set. The same analogy is repeated for 2σ (95%) and 3σ (99.7%), this was implemented for each of the 15 values of the output independently, obtaining the respective coverage probabilities relative to the output densities, and then we estimated the mean of this 15 coverage probabilities represented in Fig. 6 for the four sets. The overall results show that the model is correctly estimating the distribution of the data, since the bars are very close to the three percentages, a little fluctuation can be seen in the 68%, where the four sets are overestimating the uncertainty of the results, meaning, we have a percentage of data bigger than 68% in 1σ of the model, so the σ of the model should be smaller, this can be seen even more for set 3.

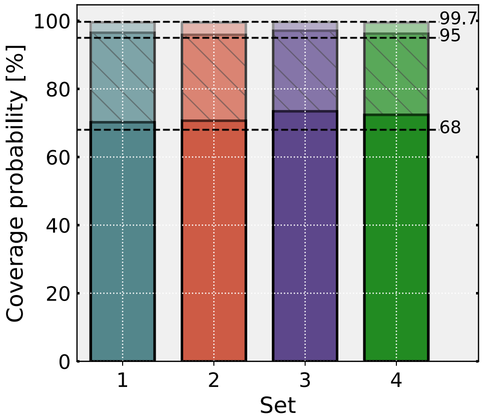

FIG. 6. Coverage probability calculated on the test set of the $v_s^2(n)$ BNNs models.

To quantify how the increase in the mock observational scattering of (R, Λ) affects the model prediction uncertainties, let us define the following quantity

$$\eta[a,b](n_k) = \frac{1}{T}\left(\sum_{i=1}^{T} \frac{\sigma_i^b(n_k) - \sigma_i^a(n_k)}{\sigma_i^a(n_k)}\right) \times 100, \qquad (23)$$

where T is the total number of EOS in the test set, and $k = 1, \ldots, 15$. This quantity defines the percentage uncertainty deviation between models b and a at density n_k. Figure 7 shows the results, where we plotted four different

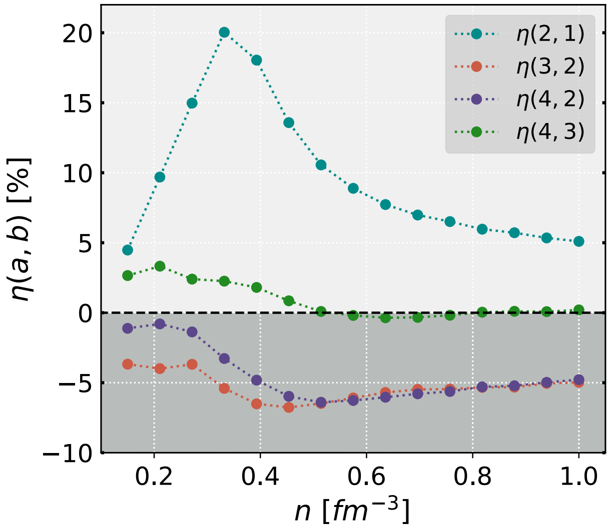

FIG. 7. Prediction uncertainty deviation $\eta[a,b]$ between the $v_s^2(n)$ BNN models a and b (see text for details).

comparisons. The first conclusion, looking at the result for $\eta[2,1]$ (cyan), is that the prediction uncertainty increased when we consider the BNN model of dataset 1 compared with the one from dataset 2. Furthermore, the $\eta[2,1]$ reached the maximum value of 20% at $n_4 = 0.33\ \mathrm{fm}^{-3}$. In other words, when the (synthetic) observational data scattering doubles, from $(\sigma_M = 0.05M_\odot, \sigma_R = 0.15\ \mathrm{km})$ to $(\sigma_M = 0.1M_\odot, \sigma_R = 0.3\ \mathrm{km})$, the uncertainty increases of the order 5–20%, having the largest value at the densities where the NS radius, $R(M)$, and $v_s^2(n)$ have the highest correlation (see the Appendix for details and Ref. [65]). The region of larger sensitivity to the uncertainty of the mock observational data coincides with the density interval where the speed of sound increases steadily, and, in many agnostic approaches attains a maximum followed by a decrease or flattening at larger densities [66–68]. The nucleonic EOS that has been used to train the model also shows similar behavior for the speed of sound [19]. This behavior of the speed of sound for densities below three times saturation density is dictated by the two solar mass constraints.

The second important conclusion is the impact of adding the Λ information on the inference properties. This point is clear when analyzing the dependence and values of $\eta[3,2]$ (the difference between datasets 2 and 3 is that the latter contains information on the tidal deformability, see Table III). The negative values reflect the fact that the prediction uncertainty decreases when the tidal information is added to the training procedure—the tidal deformability is informative of the $v_s^2(n)$ of neutron star matter. The maximum value of the uncertainty decrease is 7%, occurring at $n_6 = 0.45\ \mathrm{fm}^{-3}$. Similarly, when considering $\eta[4,2]$ (blue), it also exhibits negative values throughout, indicating a decrease in uncertainty compared to dataset 2. However, beyond the seventh density, the two values become very close to each other, suggesting that no additional information is being perceived by including more dispersion on the tidal deformability. Looking at $\eta[4,3]$ and comparing it with $\eta[2,1]$, we see that for the first density, they differ very little, but from there on $\eta[2,1]$ is approximately always more than five times bigger. Noting an important consideration here, the proportion of input values being altered: datasets 3 and 4 involve changing only the uncertainty of five quantities out of the 20 input values, i.e., changing a quarter of our input vector, while in datasets 1 and 2 we have changed all the input values. So we could anticipate an at least fourfold increase in η for datasets 1 and 2, however, this percentage is the majority of times even bigger. This implies that the dispersion of the mass-radius pairs has a more significant impact on the model compared to the mass-tidal deformability pairs. By acknowledging this difference in the proportion of modified input values, we gain insight into how the model's understanding and response to dispersion changes are influenced.

B. Proton fraction

Let us now analyze the model's predictions for the proton fraction $y_p(n)$. Using a specific EOS from the test dataset (for illustration purposes), we show in Fig. 8 the models' prediction for each dataset: 1 (blue) and 2 (orange) on the upper panel and 3 (purple) and 4 (green) in the lower panel. The range of $y_p(n)$ from the train set is indicated by the gray region and the dashed gray line displays the 99.9% percentage of data/probability line. The train statistics is important to point out that the upper region between the 99.9% probability and maximum boundary lines is caused by the presence of just one *extreme* EOS.

The conclusion drawn from Fig. 8 is similar to the $v_s^2(n)$ results (see Fig. 4): it is evident that dataset 1 exhibits

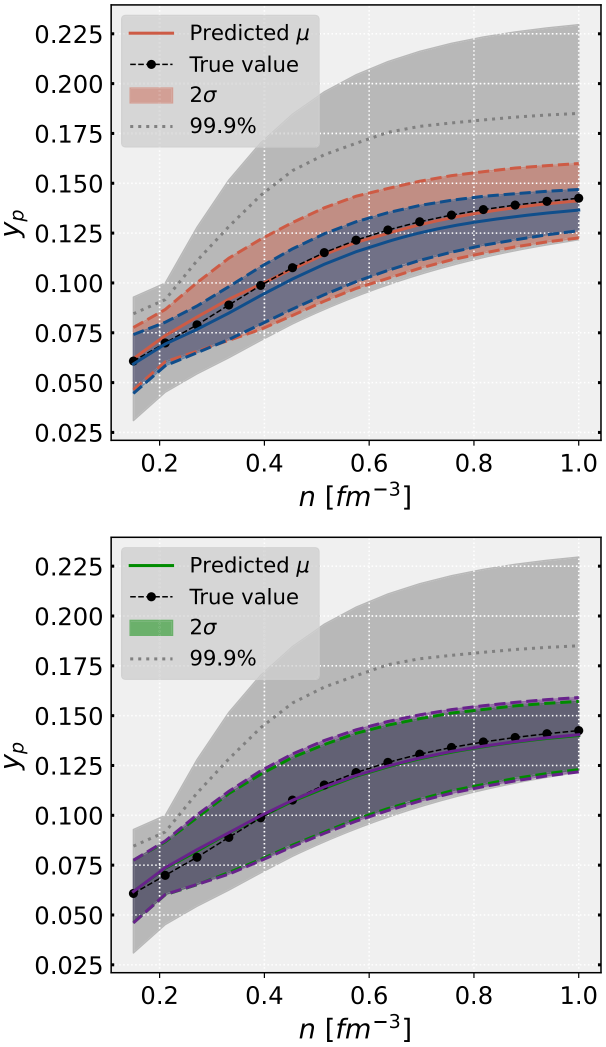

FIG. 8. The BNNs predictions for $y_p(n)$ using one EOS of the test. The models trained on datasets 1 (blue) and 2 (orange) are in the upper figure while datasets 3 (purple) and 4 (green) models are in the lower figure. The predicted mean values (solid lines) and 2σ confidence intervals are shown. The true values are shown in black dots.

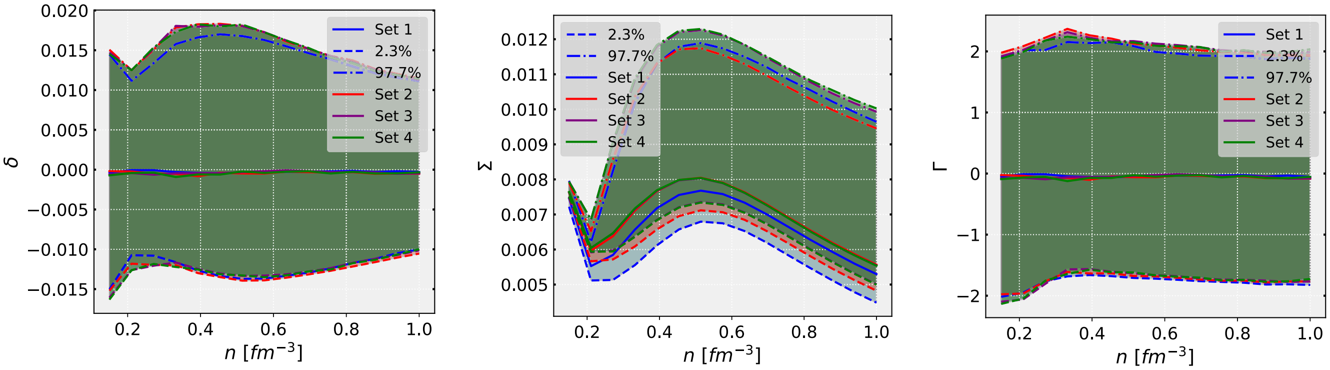

FIG. 9. Median (solid line) and the 95.4% confidence interval (dashed and dotted lines) for $\delta(n_k) = y_p(n_k) - y_p(n_k)^{\mathrm{true}}$ (left), $\Sigma(n_k) = \sigma(n_k)$ (center), and $\Gamma(n_k) = (y_p(n_k) - y_p(n_k)^{\mathrm{true}})/\sigma(n_k)$ (right).

the narrowest prediction uncertainty, whereas dataset 2 demonstrates a considerable increase in uncertainty; the prediction uncertainties are similar for datasets 3 and 4. Definite conclusions require some statistics over a set of EOS, and, for that purpose, we are going to analyze the whole test set once again.

Figure 9 shows the following quantities over the four datasets: the model residuals $\delta(n_k) = y_p(n_k) - y_p(n_k)^{\mathrm{true}}$ (left), the standard deviation $\Sigma(n_k) = \sigma(n_k)$ (center), and the normalized model residuals $\Gamma(n_k) = (y_p(n_k) - y_p(n_k)^{\mathrm{true}})/\sigma(n_k)$ (right). We calculate these quantities, at each density n_k, for each of the 2529 EOS from the test set. We see that the model has a broader residual spread around 0.4–0.5 fm^{-3} that is counterbalanced by larger standard deviation values in this region—this is a direct conclusion when looking at the normalized model residuals steadiness. In other words, the model correctly captures and models the data statistics: the BNN models capture larger prediction uncertainties in regions where $y_p(n)$ has a larger dispersion, as expected. The overall quality of the models prediction is seen in the density independence of the $\Gamma(n)$ (right panel) statistics, the models residuals are 95.4% of the times between 2σ. The insight of the train set statistics, in Fig. 10, entails interpretability for $\delta(n)$ and particularly $\Sigma(n)$ behavior, since it becomes clearer that $\sigma(n)$ of the train set has a nonmonotonic behavior, reaching a maximum value around 0.5 fm^{-3} decreasing for lower/larger densities. The coverage probability for the $y_p(n)$ models are similar to the ones obtained for $v_s^2(n)$ (see Fig. 6) and thus the models are correctly estimating the distribution of the test set data.

The dataset comparison for $\eta[a,b](n)$ [see Eq. (23)] is displayed in Fig. 11. First, we observe that $\eta[2,1]$ exhibits a behavior similar to v_s^2. However, it reaches its maximum value earlier at n_3. Interestingly, this is precisely where $R(M)$ and $y_p(n)$ demonstrate the highest correlation, as explained in the Appendix. When comparing $\eta[2,1]$ with $\eta[4,3]$, the behavior is consistent with the one observed in the speed of sound, albeit this time is even more

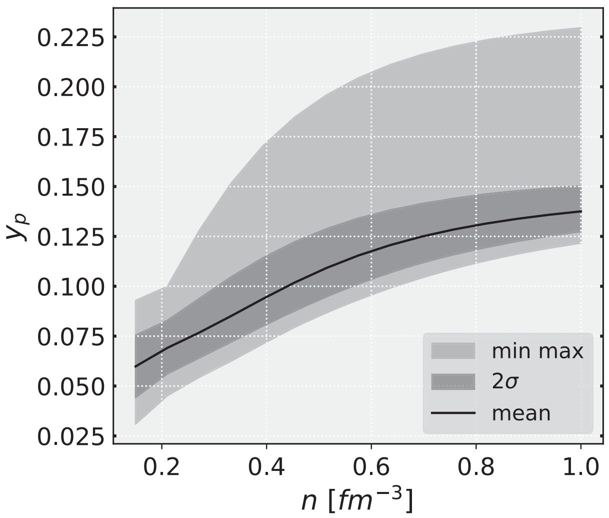

FIG. 10. Some statistics of $y_p(n)$ calculated from the train dataset.

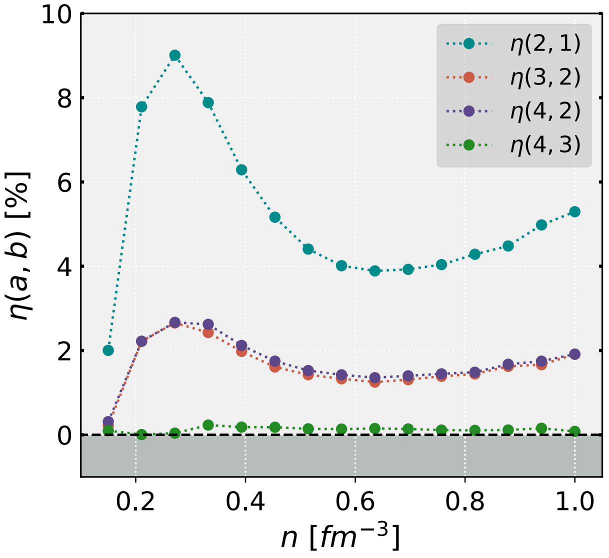

FIG. 11. Prediction uncertainty deviation $\eta[a,b]$ between the $y_p(n)$ BNN models a and b (see text for details).

outstanding, which comes back again to the correlation, that is almost zero for $\Lambda(M)$ and $y_p(n)$. Based on this observation, we can speculate that Λ does not contribute with significant information to the model, especially since this time the ratio between $\eta[4,3]$ and $\eta[2,1]$ deviates even further from the expected one-quarter proportion, that comes from the change in the input vector, than it did on the v_s^2. Increasing the model's complexity without adding substantial information can lead to increased confusion and greater uncertainty, as evident in $\eta[3,2]$ and $\eta[4,2]$. These two cases exhibit positive values of η, indicating higher uncertainty for sets 3 and 4 when compared with set 2.

C. BNNs epistemic and aleatoric uncertainties

Let us briefly analyze how the prediction uncertainty of BNNs is modeled and its different components. We are going to focus on the $y_p(n)$ BNN models. However, all the features discussed below are also seen for the $v_s^2(n)$ models. We have seen in Eq. (16) of Sec. II that the prediction variance $\hat{\boldsymbol{\sigma}}^2$ is a combination of two terms,

$$\hat{\boldsymbol{\sigma}}^2 = \hat{\boldsymbol{\sigma}}^2_{\text{alea}} + \hat{\boldsymbol{\sigma}}^2_{\text{epist}},$$

where the aleatoric uncertainty $\hat{\boldsymbol{\sigma}}^2_{\text{alea}}$ measures the mean variance of the models' ensemble, while the epistemic uncertainty $\hat{\boldsymbol{\sigma}}^2_{\text{epist}}$ measures the spread of the models around the ensemble mean $\hat{\boldsymbol{\mu}}$. The epistemic uncertainty arises from limited information or data and is encoded on the posterior probability $P(\boldsymbol{\theta}|D)$, i.e., the model distribution. On the other hand, aleatoric uncertainty is due to the inherent randomness of the dataset and is encoded into the data likelihood $P(\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta})$. While the epistemic uncertainty decreases when more data is available, the aleatoric uncertainty value does not depend on the amount of data has it is a property of the generating data process.

To analyze the proportions of both uncertainty types on the total prediction variance, we calculate the epistemic percentage as $f_{\text{epist}} = (\hat{\boldsymbol{\sigma}}^2_{\text{epist}}/\hat{\boldsymbol{\sigma}}^2) \times 100\%$. Using the BNNs models for $y_p(n)$, we show in the left panel of Fig. 12 the mean (dashed lines) and 68% confidence interval region (colored regions) of f_{epist} across the entire test (2 and 3) sets. Additionally, the right panel shows f_{epist} using BNN models trained on set 1 but with different numbers of mock observations: $n_s = 20$ and $n_s = 60$ (the value used in this work). The following conclusions can be drawn: (i) the prediction variance $\hat{\boldsymbol{\sigma}}^2$ is composed mainly of aleatoric uncertainty (left panel), around 95%, and it is due to the already high number of mock observations $n_s = 60$; (ii) the right panel shows that decreasing the number of mock observations n_s, and thus the total number of training points, increases the epistemic uncertainty. The epistemic uncertainty converges to zero when the data points go to infinity; and (iii) left panel also shows that f_{epist} is smaller for set 2 (orange) than set 3 (purple) because the input dimensions increase from 10 to 20, which is reflected on the posterior $P(\boldsymbol{\theta}|D)$. Lastly, let us argue why the epistemic uncertainty is larger at densities 0.2–0.4 fm^{-3}. When constructing the predicting ensemble, $P(\boldsymbol{y}^*|\boldsymbol{x}^*,D) = \frac{1}{N}\sum_{n=1}^{N} P(\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta}^{(n)})$, by sampling from the variational posterior, $\boldsymbol{\theta}^{(n)} \sim q_\phi(\boldsymbol{\theta})$, the density points n_k with larger correlation with $y_p(n)$ are much more sensitive to model sampling than other density points where correlations are much weaker.

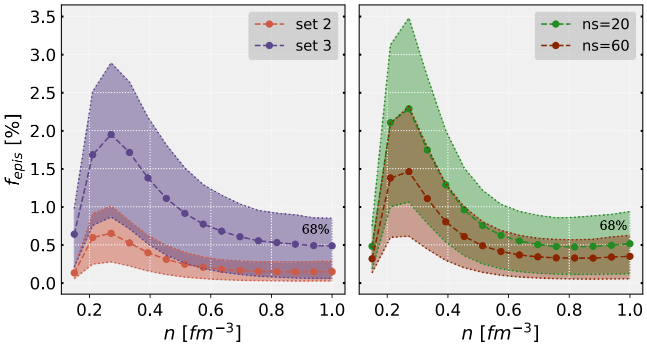

FIG. 12. The pdf of $f_{\text{epist}} = (\hat{\boldsymbol{\sigma}}^2_{\text{epist}}/\hat{\boldsymbol{\sigma}}^2) \times 100\%$ for $y_p(n)$ BNN models in sets 2 and 3 (left) and for set 1 model but trained in different training datasets with different number of mock observations n_s (right).

VI. PREDICTION FOR THE DD2 NUCLEAR MODEL

As a final test, we applied the BNN model (trained on set 1, see Table III) to a nuclear model with different properties from the ones used to train, in particular, obtained within a different microscopic description of nuclear matter. We select the DD2 model which is a generalized RMF model with density-dependent couplings [69], which has been calibrated to describe properties of finite nuclei. One key difference between the DD2 and the RMF family we used to generate the set of EOS consists of the high density behavior of the symmetry energy. In DD2 model, the coupling to the ρ meson that defines the isovector channel of the EOS goes to zero at sufficiently high densities, favoring very asymmetric matter. One of the main consequences is that nucleonic direct Urca processes inside NS are not predicted by DD2 [70,71]. Another noticeable difference between the DD2 class of models and the class of models used to train the BNN is the behavior of the speed of sound with density: for DD2 like models the speed of sound increases monotonically, although it remains always well below c, while for the class of models used to train BNN the speed of sound flattens or even decreases above $\sim 3\rho_0$. These two differences will be reflected in the performance of the BNN model.

After selecting the DD2 EOS, following the statistical procedure described in Sec. IV, we generated one mock

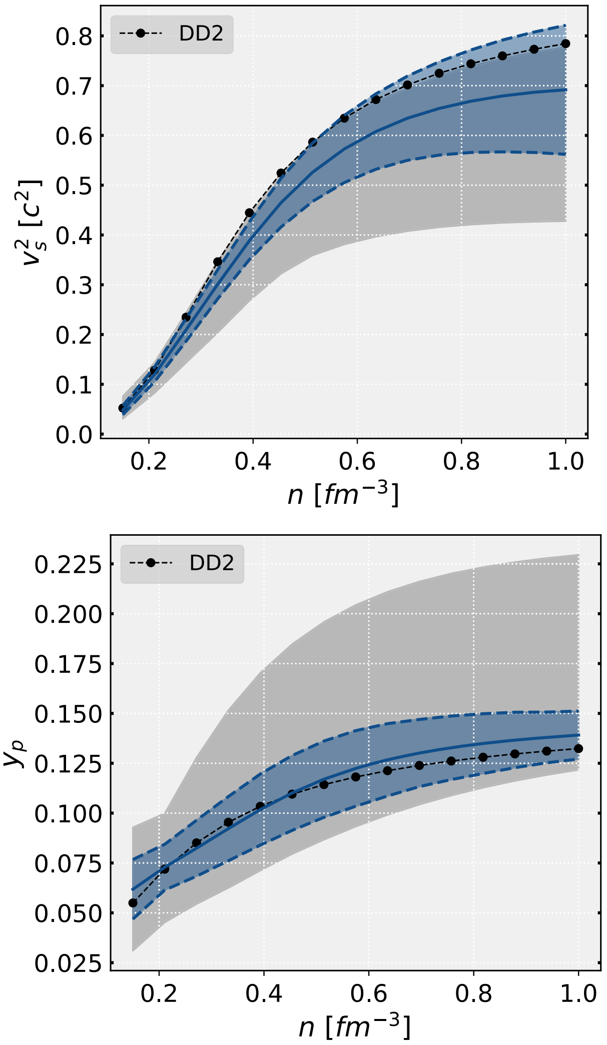

FIG. 13. The BNN model predictions, v_s^2 (upper) and y_p (lower), for one mock observation ($n_s = 1$) of the DD2 EOS, the blue area represents the 95% confidence interval, and the solid line the mean.

observation ($n_s = 1$) using the dataset 1 properties, which is the one with lower σ_R and does not contain the information about Λ. The BNN model predictions for the speed of sound (top panel) and proton fraction (lower panel) are shown in Fig. 13. Despite the DD2 lying outside the training values (gray region) for the speed of sound, the model prediction uncertainty extends beyond the training maximum values and almost contains completely the DD2 results. The $y_p(n)$ prediction is quite good, being the prediction of the mean value close to the real one. The DD2 proton fraction reflects the property described above concerning its favoring large neutron-proton asymmetries due to the behavior of the isovector channel. However, our BNN model was able to capture this behavior. Despite the results being quite compelling, there are some crucial points that we would like to point out. During the above test stage, we generated just one mock observation ($n_s = 1$) from the DD2 $M(R)$ curve to simulate a real case scenario, where a very limited number of NS observations are accessible. Being the sampling procedure of generating one mock observation ($n_s = 1$), i.e., five $M_i(R_i)$ values, a random process, implies that different samples will originate different predictions [$v_s^2(n)$ is much more sensitive than $y_p(n)$, since the DD2 target is completely inside the training values region]. This is a somehow expected behavior since we are trying to characterize the whole $M(R)$ with only five random (M, R) values. While a given sample may be sufficient to inform the general dependence of the $M(R)$ (like the one we generated), others might be as well almost uninformative of the actual $M(R)$ curve, i.e., a sample where all five mock observations $M_i(R_i)$ cluster around the same M value. This is a general problem that shows up regardless of the inference model or framework: inferring the EOS from a very limited number of NS observations. The BNN performance assessment for the DD2 EOS would get more reliable as the number of points $M_i(R_i)$, which compose each mock observation (five in the present work), increase since a random sample, in that case, would be much more informative of the true $M(R)$ curve.

VII. CONCLUSIONS

We have explored BNNs, which is a probabilistic machine learning model, to predict the proton fraction and speed of sound of neutron star matter from a set of NS mock observations. This method is based upon the usual neural networks but with the crucial advantage of attributing an uncertainty measurement to its predictions. Our EOS dataset was generated from a relativistic mean field approach through a Bayesian framework, where constraints from nuclear matter properties and NS observations were applied. The choice of a specific microscopic nuclear model, instead of a more flexible EOS parametrization, as the ones discussed for instance in [72], is justified because we want to analyze the possibility of inferring the neutron star composition, specifically, the proton fraction, from NS observations. From the set of 25287 EOS, four different mock observational sets, simulating four different scenarios of mock observational uncertainties, were generated. Two of them are only composed of $M(R)$ simulated observations and the other two have also information regarding $\Lambda(M)$. In the end, eight different BNNs were trained to predict the $v_s^2(n)$ and $y_p(n)$ in each of the four datasets.

With this study, we have shown that using BNNs, the measurements of the mass and radius of five neutron stars allow us to recover information from the equation of state of nuclear matter with associated uncertainty, not only for a quantity that is more connected with the isoscalar behavior of the EOS, the speed of sound, but also for the proton fraction, a property that is determined by the isovector behavior of the EOS. In several recent works, the attempt to determine the proton fraction from the mass radius measurements was unsuccessful [21–23]. In all these descriptions a polynomial expansion of the EOS until the third or fourth order has been considered. In [23], this was attributed to the existence of multiple solutions. The authors of [22] identify the correlations among higher-order parameters as a difficulty. The BNN approach allows the model to learn the full density dependence of the EOS avoiding the short comes of the density expansion with a finite number of terms. It was shown that the uncertainty associated with the predicted quantities is particularly sensitive to the precision of the observational data if some kind of correlation exists between the data and the property that is being calculated. For the speed of sound this was reflected in a larger sensitivity for densities below three times saturation density, where the NS radius is strongly correlated with the speed of sound as discussed in [65]. It was also shown that adding extra observational mock data, in particular, the tidal deformability, could decrease the uncertainty associated with the prediction but not always. There was a clear improvement for the speed of sound but not for the proton fraction. Too scattered data does not bring an improvement on the uncertainty determination due to the increase of the complexity of the model compared with the quality of the data. It is important to point out that the improvement attained with the tidal deformability data with a smaller uncertainty worked for the speed of sound because it shows a correlation with the tidal deformability for densities of the order of twice saturation density similar to the one with the radius. This correlation does not exist between the proton fraction and the tidal deformability so no improvement in the proton fraction prediction was attained when introducing the tidal deformability observation. The proton fraction has shown some sensitivity at twice saturation density to the radius uncertainty and this can be traced back to the existing correlation of low mass star radius with the symmetry energy slope [73], a quantity that strongly determines the proton fraction. This correlation weakens quickly with the increase of the NS mass, and is much weaker with the tidal deformability. We have also tested the BNN model with a mock measurement obtained from the DD2 EOS generated with a microscopic framework different from the one used to generate the EOS used to train the BNN model. The results have confirmed the validity of the model and its predicting power.

We have been very conservative concerning the uncertainties attached to the observations. In the future, observatories such as STROBE-X [16] and eXTP [14] may give us radius measurements with uncertainties as small as 2%–5% and this will improve the predictions as demonstrated in the present study.

There are several potential paths for further improvement and exploration in this work. One possibility is to extend the analysis to include other properties of neutron stars and investigate their relationship with observable quantities. Another possibility for improvement as discussed in the results obtained for the DD2 model, is increasing the number of observable pairs used as input can enhance the model's performance. However, it is worth noting that in the case of BNNs, expanding the number of pairs introduces a greater increase in the model parameters compared to traditional architectures, which is why we used a lesser amount of pairs compared with previous articles using conventional neural networks, as demonstrated in studies like [25] and related articles. Furthermore, for the stochastic model, it would be interesting to improve the prior as mentioned in Sec. II A.

ACKNOWLEDGMENTS

This work was partially supported by national funds from FCT (Fundação para a Ciência e a Tecnologia, I.P, Portugal) under Projects No. UIDP/04564/2020, No. UIDB/04564/2020, and No. 2022.06460.PTDC.

APPENDIX: CORRELATION BETWEEN NS PROPERTIES AND EOS

Figure 14 shows the Pearson correlation coefficient between $v_s^2(n)$ and $R(M)$ (left panel) and $\Lambda(M)$ (right panel) for specific NS masses (colors) and the average value, by considering $M/M_\odot \in [1, 2.2]$. The same is performed for the proton fraction in Fig. 15. The Pearson correlation is calculated as $\mathrm{Corr}(a,b) = \mathrm{Cov}(a,b)/(\sigma_a \sigma_b)$, where a consists of v_s^2 and y_p and b of R and Λ. Note, however, that this correlation measure is only sensitive to linear dependencies, and higher order ones can be missed. These correlations have been discussed in [65].

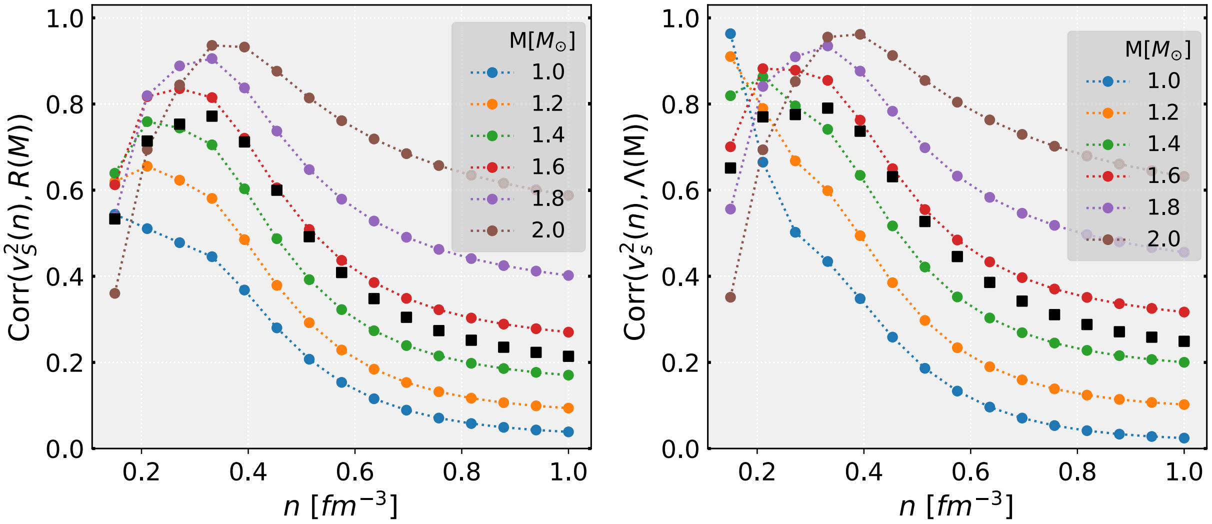

FIG. 14. Correlation between $v_s^2(n)$ and $R(M)$ (left) or $\Lambda(M)$ (right) for fixed mass values. In black squares we show the mean correlation value for $M/M_\odot \in [1, 2.2]$.

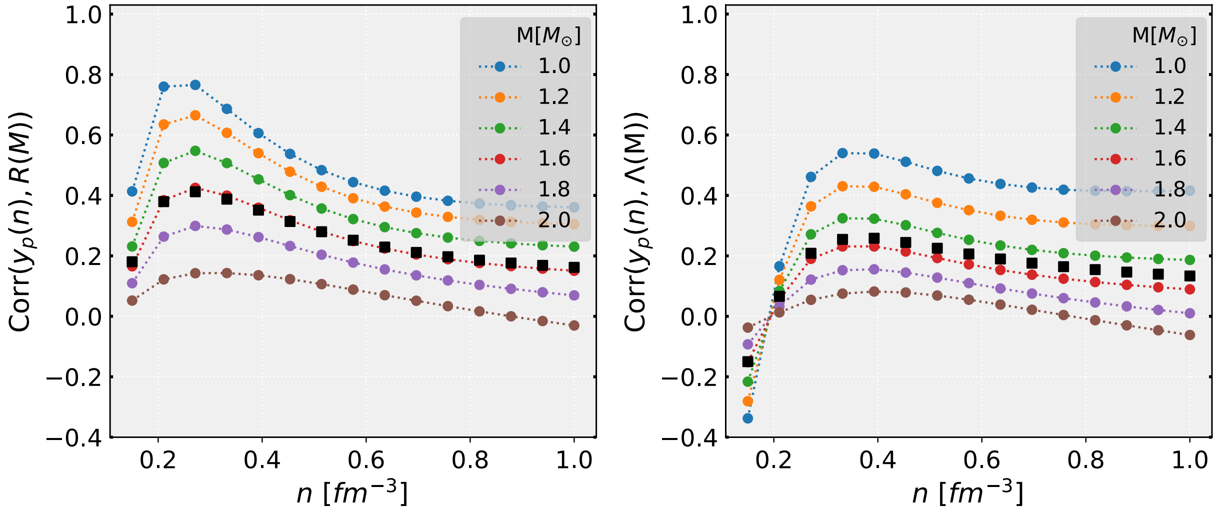

FIG. 15. Correlation between y_p and $R(M)$ (left) or $\Lambda(M)$ (right) for fixed mass values. In black squares we show the mean correlation value for $M/M_\odot \in [1, 2.2]$.

[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 *et al.*, Astrophys. J. Lett. **918**, L27 (2021).

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

[13] G. Raaijmakers, S. K. Greif, K. Hebeler, T. Hinderer, S. Nissanke, A. Schwenk, T. E. Riley, A. L. Watts, J. M. Lattimer, and W. C. G. 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), arXiv:1903.03035.

[17] A. Watts *et al.*, Proc. Sci. AASKA14 (**2015**) 043 [arXiv:1501.00042].

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

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

[20] R. Essick, I. Tews, P. Landry, S. Reddy, and D. E. Holz, Phys. Rev. C **102**, 055803 (2020).

[21] P. B. de Tovar, M. Ferreira, and C. Providência, Phys. Rev. D **104**, 123036 (2021).

[22] S. M. A. Imam, N. K. Patra, C. Mondal, T. Malik, and B. K. Agrawal, Phys. Rev. C **105**, 015806 (2022).

[23] C. Mondal and F. Gulminelli, Phys. Rev. D **105**, 083016 (2022).

[24] R. Essick, I. Tews, P. Landry, and A. Schwenk, Phys. Rev. Lett. **127**, 192701 (2021).

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

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

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

[28] S. Soma, L. Wang, S. Shi, H. Stöcker, and K. Zhou, Phys. Rev. D **107**, 083028 (2023).

[29] S. Soma, L. Wang, S. Shi, H. Stöcker, and K. Zhou, J. Cosmol. Astropart. Phys. 08 (2022) 071.

[30] S. Chatterjee, H. Sudhakaran, and R. Mallick, arXiv: 2302.13648.

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

[32] F. Morawski and M. Bejger, Phys. Rev. C **106**, 065802 (2022).

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

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

[35] M.-Z. Han, S.-P. Tang, and Y.-Z. Fan, Astrophys. J. **950**, 77 (2023).

[36] P. G. Krastev, Symmetry **15**, 1123 (2023).

[37] S. Traversi and P. Char, Astrophys. J. **905**, 9 (2020).

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

[39] M. Ferreira, V. Carvalho, and C. Providência, Phys. Rev. D **106**, 103023 (2022).

[40] G. Gonçalves, M. Ferreira, J. Aveiro, A. Onofre, F. F. Freitas, C. Providência, and J. A. Font, arXiv:2210.08382.

[41] D. Farrell, P. Baldi, J. Ott, A. Ghosh, A. W. Steiner, A. Kavitkar, L. Lindblom, D. Whiteson, and F. Weber, J. Cosmol. Astropart. Phys. 02 (2023) 016.

[42] A. Thete, K. Banerjee, and T. Malik, arXiv:2208.13163.

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

[44] W. Zhou, J. Hu, Y. Zhang, and H. Shen, Astrophys. J. **950**, 186 (2023).

[45] D. J. MacKay, Neural Comput. **4**, 448 (1992).

[46] S. Bollweg, M. Haußmann, G. Kasieczka, M. Luchmann, T. Plehn, and J. Thompson, SciPost Phys. **8**, 006 (2020).

[47] Y.-C. Lin and J.-H. P. Wu, Phys. Rev. D **103**, 063034 (2021).

[48] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, and M. Bennamoun, IEEE Comput. Intell. Mag. **17**, 29 (2022).

[49] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, in *International Conference on Machine Learning* (PMLR, Lille, France, 2015), pp. 1613–1622.

[50] A. Olivier, M. D. Shields, and L. Graham-Brady, Comput. Methods Appl. Mech. Eng. **386**, 114079 (2021).

[51] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, D. B. Rubin, J. Carlin, H. Stern, D. Rubin, and D. Dunson, *Bayesian Data Analysis Third edition* (CRC Press, Boca Raton, Florida, 2013).

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

[53] S. Typel and H. H. Wolter, Nucl. Phys. **A656**, 331 (1999).

[54] 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).

[55] S. Shlomo, V. M. Kolomietz, and G. Colò, Eur. Phys. J. A **30**, 23 (2006).

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

[57] R. Essick, P. Landry, A. Schwenk, and I. Tews, Phys. Rev. C **104**, 065804 (2021).

[58] 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*, American Institute of Physics Conference Series Vol. 735, edited by R. Fischer, R. Preuss, and U. V. Toussaint (American Institute of Physics, New York, 2004), pp. 395–405.

[59] J. Buchner, A. Georgakakis, K. Nandra, L. Hsu, C. Rangel, M. Brightman, A. Merloni, M. Salvato, J. Donley, and D. Kocevski, Astron. Astrophys. **564**, A125 (2014).

[60] J. Buchner, Stat. Surv. **17**, 169 (2023).

[61] D. P. Kingma and J. Ba, arXiv:1412.6980.

[62] S. J. Reddi, S. Kale, and S. Kumar, arXiv:1904.09237.

[63] M. Abadi *et al.*, TensorFlow: Large-scale machine learning on heterogeneous systems (2015), software available from https://www.tensorflow.org/.

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

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

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

[67] T. Gorda, O. Komoltsev, and A. Kurkela, Astrophys. J. **950**, 107 (2023).

[68] R. Somasundaram, I. Tews, and J. Margueron, Phys. Rev. C **107**, 025801 (2023).

[69] S. Typel, G. Ropke, T. Klahn, D. Blaschke, and H. H. Wolter, Phys. Rev. C **81**, 015803 (2010).

[70] M. Fortin, C. Providencia, A. R. Raduta, F. Gulminelli, J. L. Zdunik, P. Haensel, and M. Bejger, Phys. Rev. C **94**, 035804 (2016).

[71] M. Fortin, A. R. Raduta, S. Avancini, and C. Providência, Phys. Rev. D **103**, 083004 (2021).

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

[73] N. Alam, B. K. Agrawal, M. Fortin, H. Pais, C. Providência, A. R. Raduta, and A. Sulaksono, Phys. Rev. C **94**, 052801 (2016).