Toward Accelerated Nuclear-physics Parameter Estimation from Binary Neutron Star Mergers: Emulators for the Tolman–Oppenheimer–Volkoff Equations

Brendan T. Reed[1] , Rahul Somasundaram[1,2] , Soumi De[1] , Cassandra L. Armstrong[3] , Pablo Giuliani[4] ,
Collin Capano[2,5] , Duncan A. Brown[2] , and Ingo Tews[1]
[1] Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA
[2] Department of Physics, Syracuse University, Syracuse, NY 13244, USA
[3] Intelligence and Space Research Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA
[4] Facility for Rare Isotope Beams, Michigan State University, East Lansing, MI 48824, USA
[5] Physics Department, University of Massachusetts Dartmouth, North Dartmouth, MA 02747, USA
Received 2024 June 11; revised 2024 August 5; accepted 2024 August 22; published 2024 October 17

Abstract

Gravitational-wave observations of binary neutron-star (BNS) mergers have the potential to revolutionize our understanding of the nuclear equation of state (EOS) and the fundamental interactions that determine its properties. However, Bayesian parameter estimation frameworks do not typically sample over microscopic nuclear-physics parameters that determine the EOS. One of the major hurdles in doing so is the computational cost involved in solving the neutron-star structure equations, known as the Tolman–Oppenheimer–Volkoff (TOV) equations. In this paper, we explore approaches to emulating solutions for the TOV equations: multilayer perceptrons (MLPs), Gaussian processes, and a data-driven variant of the reduced basis method (RBM). We implement these emulators for three different parameterizations of the nuclear EOS, each with a different degree of complexity represented by the number of model parameters. We find that our MLP-based emulators are generally more accurate than the other two algorithms, whereas the RBM results in the largest speedup with respect to the full high-fidelity TOV solver. We employ these emulators for a simple parameter inference using a potentially loud BNS observation and show that the posteriors predicted by our emulators are in excellent agreement with those obtained from the full TOV solver.

Unified Astronomy Thesaurus concepts: Neutron stars (1108); Nuclear astrophysics (1129); Nuclear physics (2077); Gravitational waves (678); Computational methods (1965)

1. Introduction

Multimessenger observations of neutron stars (NSs), such as the detection of the binary neutron-star (BNS) merger GW170817 (B. P. Abbott et al. 2017a, 2017b), NS mass observations (P. Demorest et al. 2010; J. Antoniadis et al. 2013; H. T. Cromartie et al. 2019; E. Fonseca et al. 2021), and observations of NSs by NASA's Neutron Star Interior Composition Explorer (M. C. Miller et al. 2019, 2021; T. E. Riley et al. 2019, 2021), have given us fascinating new insights into the equation of state (EOS) of the densest matter in the Universe (E. Annala et al. 2018; T. Malik et al. 2018; I. Tews et al. 2018b; G. Raaijmakers et al. 2019; C. D. Capano et al. 2020; T. Dietrich et al. 2020; R. Essick et al. 2020; H. Güven et al. 2020; P. Landry et al. 2020; M. Al-Mamun et al. 2021; S. Huth et al. 2022; M. Breschi et al. 2024; H. Koehn et al. 2024). Nevertheless, several questions regarding the properties of dense matter (J. M. Lattimer & M. Prakash 2004), such as the possibility of a QCD phase transition in the inner core of NSs (E. Annala et al. 2020; R. Somasundaram et al. 2023), remain unanswered. The advent of the next-generation of gravitational-wave (GW) detectors, such as Cosmic Explorer (D. Reitze et al. 2019; M. Evans et al. 2021) and the Einstein Telescope (M. Punturo et al. 2010; M. Branchesi et al. 2023), is expected to dramatically improve our constraints on the EOS, thereby providing us with key information to answer these open

questions in nuclear astrophysics (D. Finstad et al. 2023; H. Rose et al. 2023; A. Bandopadhyay et al. 2024).

Prospects for measuring the EOS with GWs have been studied in the literature, in the context of both present (B. P. Abbott et al. 2018; E. Annala et al. 2018; S. De et al. 2018; T. Malik et al. 2018; C. D. Capano et al. 2020; R. Essick et al. 2020; N. Kunert et al. 2022) and next-generation (P. Landry & K. Chakravarti 2022; F. Iacovelli et al. 2023; H. Rose et al. 2023; A. Bandopadhyay et al. 2024; A. Prakash et al. 2024; K. Walker et al. 2024) GW detectors. However, these studies typically do not sample over microphysical nuclear-physics parameters directly, i.e., the Tolman–Oppenheimer–Volkoff (TOV) equations are not evaluated for each sample in the Bayesian inference framework but typically beforehand. Then, one samples over these pregenerated EOS sets. However, an "on-the-fly" generation of EOS from its microphysical parameters is required to comprehensively probe the underlying nuclear physics from astrophysical observations and directly constrain the fundamental interactions among the degrees of freedom. Instead, hierarchical inference (F. Hernandez Vivanco et al. 2019; P. Landry & R. Essick 2019; R. Essick et al. 2020; K. Walker et al. 2024) is often used to first obtain the marginalized likelihood on the NS component masses and tidal deformabilities, which is then integrated over different EOS curves to ultimately yield constraints on the EOS. Alternatively, some studies (C. D. Capano et al. 2020; T. Dietrich et al. 2020; P. T. H. Pang et al. 2023) pretabulate large numbers of EOSs and sample over an EOS index during the parameter inference run. Both approaches have shortcomings. The former does not incorporate density-dependent theoretical uncertainties in the EOS models when the parameter inference is

performed. The latter approach does not suffer from this drawback but might be unsatisfactory for calculating posterior distributions over EOS parameters. This is because this approach maps one sampling parameter—the EOS index—to several nuclear-physics parameters.

To overcome these problems, it is important to sample over the nuclear-physics parameters simultaneously with the other binary parameters. This implies an "on-the-fly" generation of the EOS and its resulting global NS properties. Hence, every likelihood evaluation in such a Bayesian inference run requires solving the TOV equations (J. R. Oppenheimer & G. M. Volkoff 1939). The TOV equations are the NS structure equations that relate the EOS—and hence, the nuclear-physics parameters—to astrophysical observables such as NS radii R and tidal deformabilities Λ (E. E. Flanagan & T. Hinderer 2008; T. Hinderer 2008; T. Hinderer et al. 2010; P. Landry & E. Poisson 2014) as a function of mass M. A typical parameter inference run for a BNS event requires $\approx 10^7$ likelihood evaluations (J. Veitch et al. 2015). Therefore, given that a single solution to the TOV equations takes up to a few seconds (when also solving for Λ), employing the full high-fidelity TOV solver is prohibitively expensive. Furthermore, this problem will be exacerbated in the era of third-generation GW detectors, which are expected to detect $\mathcal{O}(1000)$ high-precision BNS events per year (M. Evans et al. 2021). This large number of observations will provide unparalleled information to constrain nuclear-physics parameters if these can be sampled over directly. Finally, rapid parameter estimation from GW signals will aid efficient electromagnetic follow-up (B. Margalit & B. D. Metzger 2019).

So far, EOS parameters were directly sampled over only for a single event, GW170817 (B. P. Abbott et al. 2018), using a relatively simple spectral decomposition model for the EOS (L. Lindblom 2010). Due to the abovementioned computational expense, sampling over EOS parameters "on-the-fly" has not been attempted for more complicated EOS models whose parameters are motivated by nuclear theory and experiment. Here, we present a novel approach to this problem by employing emulators (P. Canizares et al. 2015; R. Smith et al. 2016; E. Bonilla et al. 2022), i.e., algorithms that mimic solutions to the TOV equations but at a fraction of their computational cost. Recently, such emulators have been applied to NS structure equations (M.-Z. Han et al. 2021; S. Soma et al. 2022; I. Liodis et al. 2024; J. McGinn et al. 2024; P. Tiwari & A. Pai 2024; G. Ventagli & I. D. Saltas 2024). However, these works have been limited to emulating NS radii and have focused on emulators based on neural networks. Instead, we develop three different approaches to TOV emulators: a neural network–based emulation using multilayer perceptrons (MLPs), Gaussian processes (GPs), and a data-driven variant of the reduced basis method (RBM). While our algorithms can be used to emulate any global NS property, in this paper, we focus on emulating the NS tidal deformability Λ (T. Hinderer et al. 2010) because it is directly relevant for GW observations of binary NS mergers.

The paper is organized as follows. In Section 2, we detail the construction of our three parametric EOS models and of both training and validation sets for the emulators. In Section 3, we lay out our implementation of the three emulators: MLP (Section 3.1), GP (Section 3.2), and RBM (Section 3.3). We also evaluate the performance of our emulators by validating them against a large set of pregenerated EOS samples.

Section 4 compares the three emulators, considering both their accuracy and speed. Finally, in Section 5, we study the ability of our emulators to obtain accurate probability density functions (PDFs) from a typical BNS event that could be detected by the next-generation of GW detectors. Our conclusions are presented in Section 6.

2. EOS Models and Data Generation

Models for EOS that are employed in GW inference analyses need to be versatile enough to provide sufficiently broad priors on astrophysical NS observables and sufficiently flexible so that they can be connected in a straightforward manner to experimental and theoretical nuclear-physics inputs. Common examples of such EOS models are the piecewise polytrope model (J. S. Read et al. 2009; K. Hebeler et al. 2013), the spectral decomposition approach (L. Lindblom 2010), GP (R. Essick et al. 2020; P. Landry et al. 2020), and the speed-of-sound model (I. Tews et al. 2018a; S. K. Greif et al. 2019). Here, we employ the EOS modeling approach recently used in H. Koehn et al. (2024) and O. Komoltsev et al. (2023).

In this approach, the outer core of NSs is assumed to be composed of only nucleonic degrees of freedom. The EOS in this part is described by the metamodel developed in J. Margueron et al. (2018). The metamodel is an extremely flexible density-functional approach whose model parameters are the so-called nuclear empirical parameters (NEPs; J. Margueron et al. 2018) that govern the behavior of the EOS of pure neutron matter and symmetric nuclear matter. The NEPs are given by taking the usual expansion of the nuclear matter EOS in powers of the proton–neutron asymmetry,

$$E(n, \alpha) = \mathcal{E}(n) + \alpha^2 S_2(n) + \mathcal{O}(\alpha^4), \qquad (1)$$

where $E(n, \alpha)$ is the total energy per particle, $\mathcal{E}(n)$ is the energy of symmetric matter, $S_2(n)$ is the quadratic approximation to the symmetry energy (R. Somasundaram et al. 2021), and $\alpha = \frac{n_n - n_p}{n}$ is the proton–neutron asymmetry parameter. Furthermore, the symmetry energy $S(n)$ is defined as the difference between the energy of pure neutron matter and symmetric nuclear matter, i.e., $S(n) \equiv E(n, \alpha = 1) - \mathcal{E}(n)$. The energy of symmetric matter and the symmetry energy are then expanded around nuclear saturation density n_{sat}:

$$\mathcal{E}(n) = E_{\text{sat}} + \frac{1}{2}K_{\text{sat}}x^2 + \frac{1}{6}Q_{\text{sat}}x^3 + \frac{1}{24}Z_{\text{sat}}x^4 + ... \qquad (2)$$

$$S(n) = E_{\text{sym}} + L_{\text{sym}}x + \frac{1}{2}K_{\text{sym}}x^2 + \frac{1}{6}Q_{\text{sym}}x^3$$
$$+ \frac{1}{24}Z_{\text{sym}}x^4 + ..., \qquad (3)$$

where $x = \frac{n - n_{\text{sat}}}{3n_{\text{sat}}}$. It has been shown that the metamodel is capable of reproducing the EOS of a large number of nucleonic models via a suitable adjustment of the NEPs. Constraints from microscopic ab initio nuclear theory, e.g., from chiral effective field theory (K. Hebeler et al. 2013; I. Tews et al. 2018a; C. Drischler et al. 2019; J. Keller et al. 2023) and nuclear experiment, such as PREX (B. T. Reed et al. 2021), can be incorporated in the form of suitable priors on the NEPs (H. Koehn et al. 2024).

The description in terms of nucleons might break down at higher densities. Hence, as in H. Koehn et al. (2024), we

employ the speed-of-sound model of I. Tews et al. (2018a) to describe the EOS above a breakdown density of the metamodel that we chose to be $2n_{\rm sat}$. For the speed-of-sound model, the parameters are the squared sound speeds at discrete density points. Unlike in H. Koehn et al. (2024), we fix these density points at $3n_{\rm sat}$, $4n_{\rm sat}$, and so on. The values of the squared sound speed at these density points are bounded by 10^{-5} from below to avoid the need for special treatments when solving for the tidal deformabilities (S. Postnikov et al. 2010) and by 1 from above due to causality. The resulting density-dependent speed of sound can be integrated to obtain the pressure, energy density, and chemical potential. This can, in turn, be used as input for the TOV equations, which yields the NS mass, radius, and tidal deformability; see I. Tews et al. (2018a) and R. Somasundaram et al. (2023) for more details.

We employ our full EOS model in various degrees of complexity to train our emulators. For our one-parameter model, we fix all the NEPs: $E_{\rm sat} = -16$ MeV, $n_{\rm sat} = 0.16$ fm^{-3}, $K_{\rm sat} = 230$ MeV, $Q_{\rm sat} = Z_{\rm sat} = 0$ MeV, $E_{\rm sym} = 32$ MeV, $L_{\rm sym} = 50$ MeV, and $K_{\rm sym} = Q_{\rm sym} = Z_{\rm sym} = 0$ MeV. Hence, the EOS up to $2n_{\rm sat}$ remains unchanged. Then, we assume the speed of sound above $3n_{\rm sat}$ to be constant and use its value as the one free parameter in the model; see left panel of Figure 1. For our five-parameter model, we vary the NEPs $L_{\rm sym}$, $K_{\rm sym}$ and $K_{\rm sat}$, with the other NEPs fixed at the same values as for the one-parameter model. Additionally, the speed of sound at $3n_{\rm sat}$ and $5n_{\rm sat}$ are free parameters. In this model, the speed of sound is chosen to be constant between $3n_{\rm sat}$ and $4n_{\rm sat}$, and above $5n_{\rm sat}$. Finally, we construct a ten-parameter model, which has the same level of flexibility for the EOS below $2n_{\rm sat}$ but varies the seven sound speeds at $3n_{\rm sat}$, $4n_{\rm sat}$, ...$9n_{\rm sat}$. For all models, all grid points are connected using linear segments. The left panel of Figure 1 illustrates the construction of our EOS models by showing the sound-speed profile for two samples per model.

To train our emulators, we generate a set of 100,000 samples for each EOS model and a separate set of 100,000 samples per model for validation. The EOS samples are obtained by varying the model parameters in a uniform range. All the (squared) sound-speed parameters are sampled uniformly between 0 and 1 in units of the speed of light, whereas we use the following ranges for the NEPs: $L_{\rm sym} = [20, 150]$ MeV, $K_{\rm sym} = [-300, 100]$ MeV, and $K_{\rm sat} = [200, 300]$ MeV. For each EOS sample, we solve the TOV equations to obtain masses, radii, and tidal deformabilities. To solve the set of TOV (and deformability) differential equations, we use the LSODA adaptive integration algorithm (L. Petzold 1983) in enthalpy space as in L. Lindblom (1992). The set of equations to solve for the deformability similarly uses the enthalpy, taking the form from L. Lindblom & N. M. Indik (2014), which is adapted from T. Hinderer (2008). On an average, solving the TOV for the full mass–radius curve that includes approximately 100 NSs takes around 1 s. Including the tidal deformability increases this time to ~3.5 s, but it can be as large as ~5 s for very soft EOSs. We show 100 resulting mass–radius curves for each EOS model in the right panel of Figure 1. Finally, for both the training and validation sets, we discard the samples that do not reach $2 M_\odot$ (P. Demorest et al. 2010; J. Antoniadis et al. 2013; H. T. Cromartie et al. 2019; E. Fonseca et al. 2021), approximately 30% of each set.

3. Emulation Strategy

Since our goal is to perform Bayesian parameter inference for BNS mergers by explicitly sampling over EOS model parameters, we train our emulators to take as input the EOS model parameters and output the tidal deformabilities Λ for a sequence of NS masses. Here, we focus on emulating Λ, but our framework can be easily extended to other astrophysical observables such as the NS radius. In the following, we separately discuss the construction of our three different emulators and their performance as characterized by their validation accuracies. In Section 4, we will compare the emulators in detail regarding their accuracy and speed.

3.1. Multilayer Perceptron

The MLP is considered to be the simplest feed-forward neural network, consisting of fully connected neurons (M. A. Nielsen 2015). The number of neurons in the input layer is determined by the number of input EOS model parameters, i.e., either 1, 5, or 10, depending on the EOS model. For the one-parameter model, our MLP consists of 15 neurons in the output layer, with the activation of each neuron in the output layer corresponding to $\log_{10}(\Lambda)$ of an NS with a certain mass. Therefore, the MLP effectively predicts the Λ–M curve on a fixed mass grid, which we take to be uniformly spaced with 15 points. For the one-parameter model, the upper bound of the grid is $2 M_\odot$, whereas the lower bound is set to be $1.4 M_\odot$. This is because, below $1.4 M_\odot$, there is no significant variation in Λ with respect to the EOS parameters since the EOS below $2n_{\rm sat}$ remains unchanged; see Figure 1. Similarly, for the five- and ten-parameter models, the output mass grid consists of 30 points uniformly spaced between $1 M_\odot$ and $2 M_\odot$. We have chosen this mass range because we do not expect to detect gravitational waves from a BNS merger event outside this range.

For the one-parameter model, we use two hidden layers, each with 64 neurons. For the five- and ten-parameter models, we use five hidden layers, each containing 64 neurons. These hyperparameters of the MLP architecture were chosen in order to optimize the performance of the MLP. We found that the MLP's performance accuracy increases as a function of the number of hidden layers but saturates roughly at the chosen values. On the other hand, the number of neurons per hidden layer did not significantly impact the behavior of the MLP, and the value 64 was found to be a good choice. The activation function of all hidden neurons in our MLPs is taken to be the rectified linear unit, whereas we use the identity function for the output neurons.

To increase the accuracy of the MLP emulator, we employ the method of bagging (L. Breiman 1996). For each EOS model, we construct not one, but 100 MLPs, all with the same architecture and hyperparameters discussed above. The training of the MLPs is performed using a stochastic gradient-based optimizer, ADAM (D. P. Kingma & J. Ba 2014). Therefore, the training is a nondeterministic process that results in a variance among the predictions of the MLPs. The final prediction of the full MLP ensemble is obtained by averaging over all individual MLPs. In the following, we will refer to this emulation strategy simply as MLP. Bagging 100 individual MLPs decreases the speed of the full emulator by a factor 100. To investigate the trade-off between speed and accuracy, we also investigate the
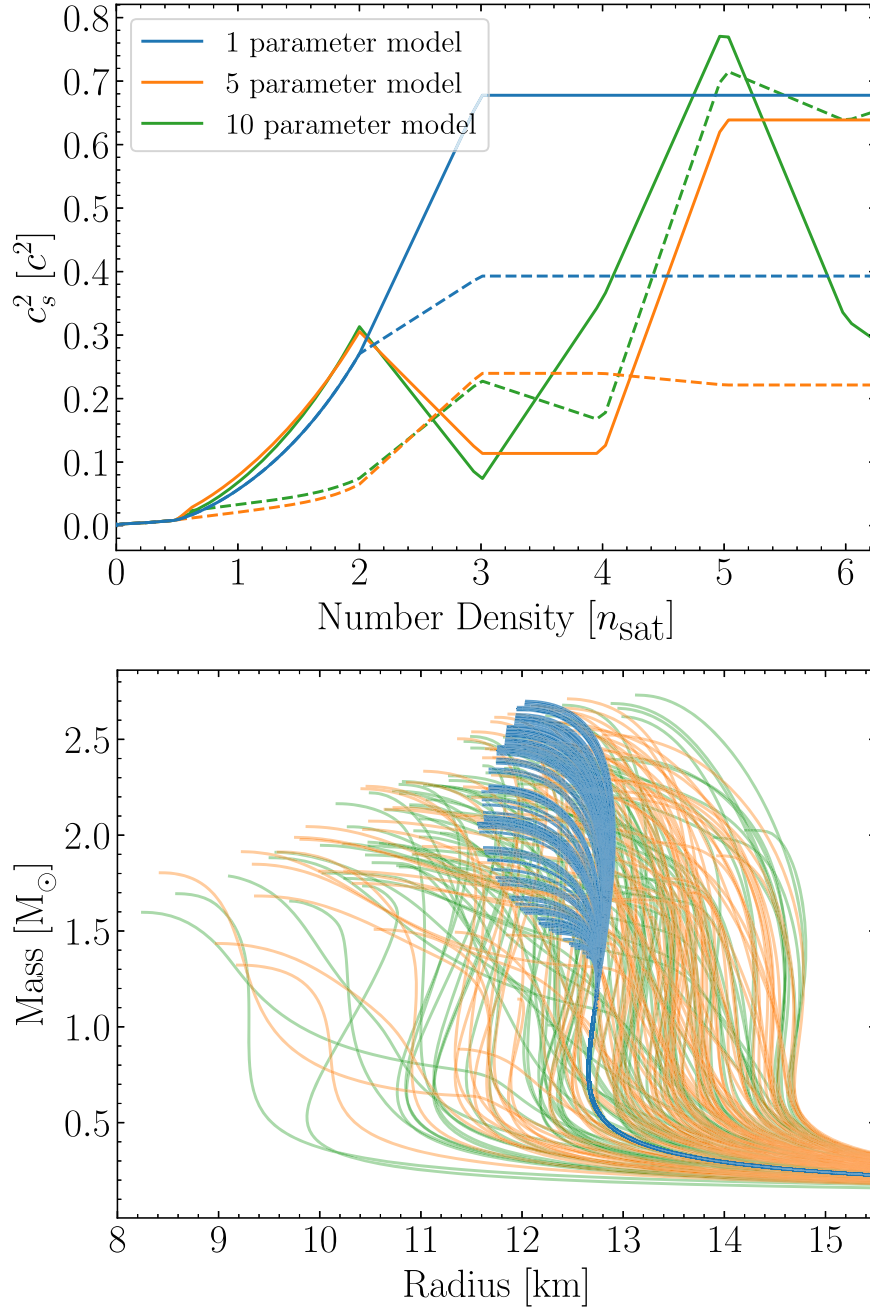
Figure 1. (Left): the density-dependent sound-speed profiles for two samples for each of our three EOS models in units of c^2, where c is the speed of light. (Right): the resulting mass–radius curves for 100 samples of each EOS model.

predictions of a single MLP in this ensemble, which we take to be the one with the lowest residual loss function. This emulator is referred to in the following as MLP*. All MLPs in the ensemble are trained using all available training samples for all EOS models. The MLPs are trained until the loss function remains constant to within 10^{-10} for 10 consecutive epochs. This typically results in a total of ≈ 100 epochs for each MLP.

Figure 2 shows the validation performance of our MLP emulator. In this section, we show results only for the five-parameter EOS model, but results are qualitatively similar for the other sets. We find that for the vast majority of validation samples, the percentage uncertainty evaluated at $1.4\,M_\odot$, $\Delta_{1.4}$, is less than 1%. The average of $\Delta_{1.4}$ over all EOSs is only 0.04%, indicating that the MLP is a faithful emulator of the full

TOV solver. The most significant outlier EOS has an uncertainty $\Delta_{1.4} \approx 15\%$, but such samples are rare, with only ≈ 10 out of $\approx 70,000$ samples exhibiting such large uncertainties. From the cross-validation plot we see that samples with larger $\Delta_{1.4}$ are softer EOSs. This is because both the training and validation data are generated by uniform sampling over the EOS model parameters. Given the nonlinear dependence of the tidal deformability on the EOS parameters, this results in a larger number of stiff EOSs, which consequently dominate the training of the MLP.

3.2. Gaussian Processes

GP regression (C. E. Rasmussen & C. K. I. Williams 2005; J. Wang 2023) offers a nonparametric, probabilistic alternative
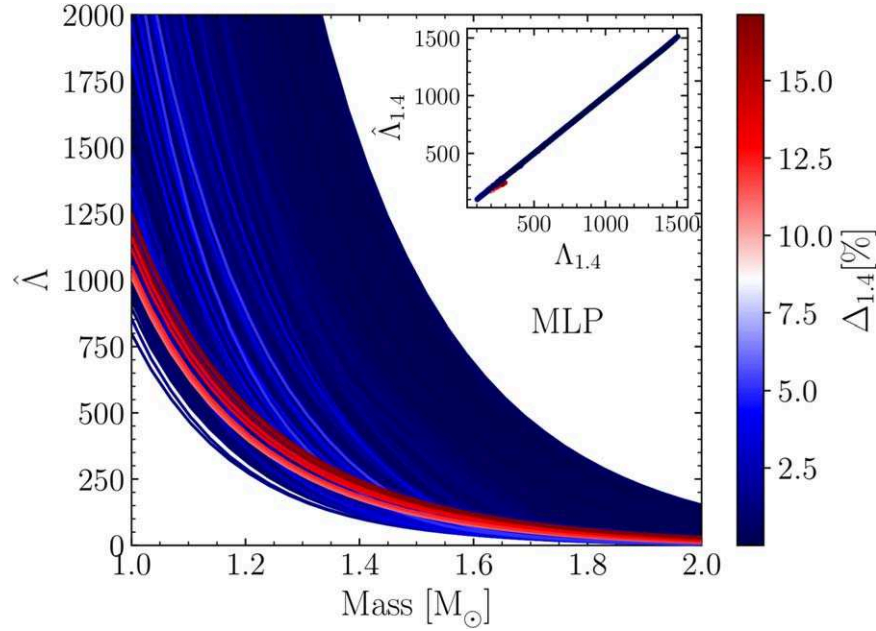
Figure 2. Validation results for the MLP for the five-parameter EOS model. We show the predicted tidal deformability $\hat{\Lambda}$ for each sample in the validation set as a function of the NS mass. The color scheme corresponds to the validation percentage uncertainty of a given sample at 1.4 $M_\odot$, $\Delta_{1.4}$. The inset depicts the cross-validation result, with the predicted tidal deformability at 1.4 $M_\odot$, $\hat{\Lambda}_{1.4}$, plotted against the corresponding true value for every validation sample.

to the MLP. Similar to the MLP, our GP emulator is designed to accept a set of EOS parameters as input and predict $\log_{10}(\Lambda)$ for a sequence of NSs on a uniform mass grid. As with the the MLP, the mass grid consists of 15 points uniformly spaced between 1.4 $M_\odot$ and 2 $M_\odot$ for the one-parameter model, whereas we use 30 points between 1 $M_\odot$ and 2 $M_\odot$ for the five- and ten-parameter models.

The prior GP is specified by the choice of the kernel, which we take to be the Matérn kernel (C. E. Rasmussen & C. K. I. Williams 2005). The Matérn kernel is a generalization of the traditional radial basis function kernel, and we found the former to generally perform better. The hyperparameters of the kernel are optimized by maximizing the log marginal likelihood, and the posterior GP is obtained using Algorithm 2.1 of C. E. Rasmussen & C. K. I. Williams (2005), as implemented in the Python package scikit-learn. In contrast to MLPs, ordinary GPs, such as those considered in this work, scale poorly with the size of the training set, and we were unable to train any GPs on more than 5000 training EOSs. There are a few approaches in the literature to address this problem, such as sparse GPs (H. Liu et al. 2020). Here, instead, we make full use of our large data set by once again employing the bagging method: instead of a single GP, we consider an ensemble of GPs, each trained on a distinct set of 5000 samples drawn from the original training set. For each GP, we use the mean of the posterior for a given set of EOS model parameters and average over the outputs of all GPs in the ensemble in order to obtain the final prediction. For the one-parameter model, we employ an ensemble of ten GPs, while for the five- and ten-parameter models, we use 20 GPs.

Figure 3 shows the validation performance of our GP emulator for the five-parameter EOS model. We find that, generally, the GP provides a good emulation of the full TOV solver, and the average $\Delta_{1.4}$ is ≈0.09%. The most significant outlier has an uncertainty $\Delta_{1.4} \approx 25\%$, and around 12 samples have such an uncertainty. Similar to the results of the MLP emulator, we see that samples with larger uncertainties tend to

be softer EOSs. While a little worse, the GP validation uncertainty is comparable to the MLP emulator and indicates that the GP can be a useful tool for emulating the TOV equations.

3.3. The Reduced-Basis Method

The RBM (A. Quarteroni et al. 2015) is a model-order reduction technique (A. Quarteroni et al. 2014) that aims at emulating a complex system by finding reduced coordinates, usually as coefficients of a small basis expansion, and creating a system of equations for them, usually through a Galerkin projection formalism (E. Bonilla et al. 2022). Even though it has been successfully applied to various problems in nuclear physics (C. Drischler et al. 2023; P. Giuliani et al. 2023; D. Odell et al. 2024; R. Somasundaram et al. 2024) and GW astronomy (S. E. Field et al. 2011; R. Smith et al. 2016), obtaining the reduced equations from a Galerkin projection is not always a viable option due to the structure of the differential equations involved. This is the case for the TOV equations we address here, since they are nonaffine in the EOS (pressure, energy density, etc.), and in the controlling EOS parameters.

To overcome this issue, we take a data-driven approach to estimate a set of reduced equations without explicitly calculating the projections (D. Figueroa et al. 2024, in preparation), as has been done in various other model-order reduction applications; see, for example ,A. M. Burohman et al. (2023). An alternative approach would rely on the collocation framework (Y. Chen et al. 2021), which can still exploit information about the operators involved to construct the reduced equations without the disadvantages of a Galerkin projection scheme. This avenue is in preparation and will soon be publicly available (A. Semposki et al. 2024, in preparation).

In the following, we describe our approach for emulating the tidal deformability $\Lambda(M)$. The emulation strategy for the radius follows the same procedure but is not discussed here. We first
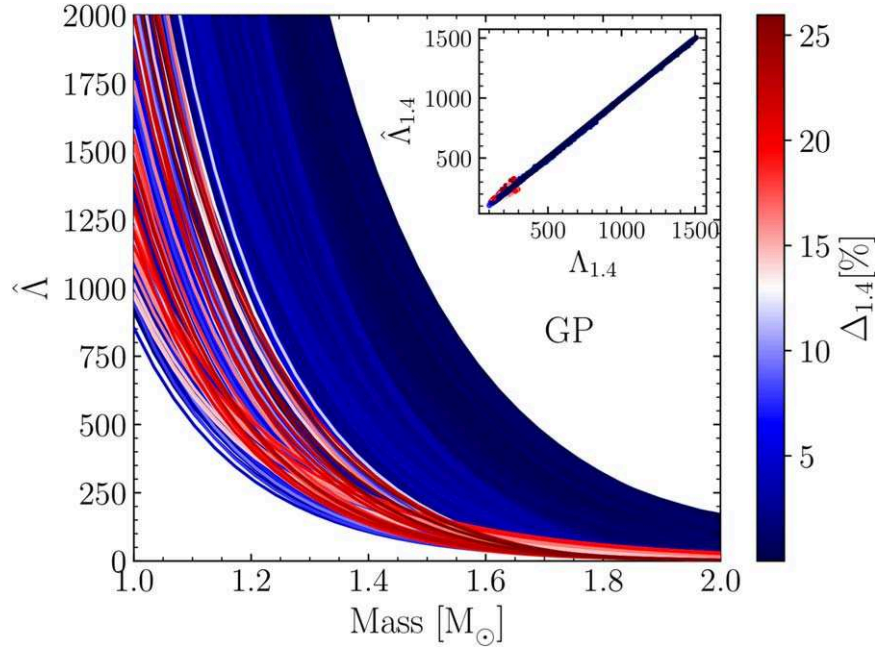
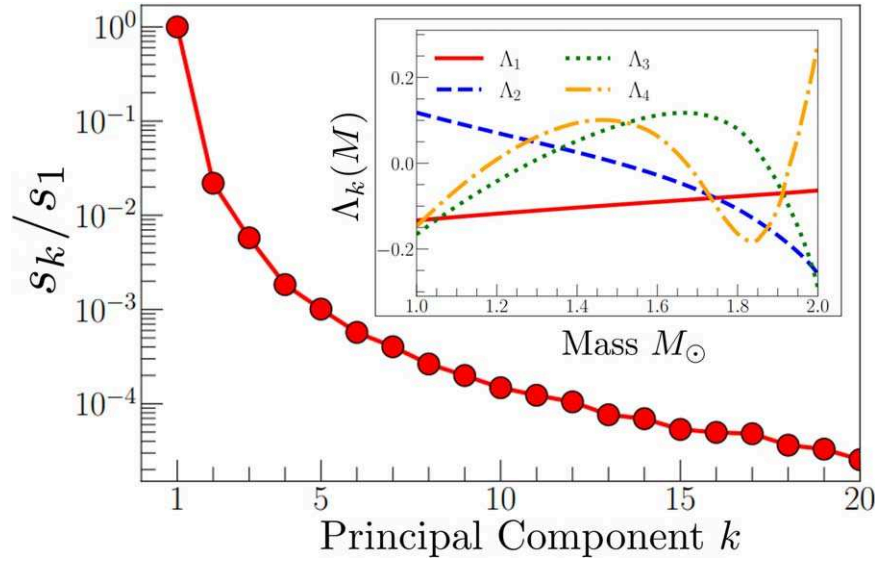**Figure 3.** Same as Figure 2 but for the GP emulator.

Figure 4. Fast decay of the singular values in the reduced basis expansion for $\log(\Lambda(M))$ in Equation (4), which signals that a good approximation can be obtained with just a few components (E. Bonilla et al. 2022). The inset shows the first four principal components, which represent the first four reduced bases $\Lambda_k(M)$.

construct a reduced basis for the $\log_{10}$ (hereafter just log) of Λ as a function of the NS mass and the EOS model parameters $\boldsymbol{\alpha}$:

$$\log(\Lambda(M; \alpha)) \approx \log(\hat{\Lambda}(M; \alpha)) = \sum_k^n a_k(\alpha)\Lambda_k(M). \quad (4)$$

Here, $\Lambda_k(M)$ represent the reduced bases, and the coefficients a_k the n represent reduced coordinates that change as we modify the parameters α. We use $\log(\Lambda)$ in order to provide smoother curves for each reduced basis, which will become important for the next step. To obtain an optimal basis $\Lambda_k(M)$, we perform a proper orthogonal decomposition (A. Quarteroni et al. 2015) and construct them as the first n principal components (or singular vectors) of a matrix of snapshots of $\log(\Lambda(M; \alpha_j))$ for $j = 1,...,N$ high-fidelity solutions of the TOV equations. The

snapshot solutions are calculated at a grid of fixed mass M values between $1.0\ M_\odot$ and $2.0\ M_\odot$. Figure 4 shows the decay of the singular values as well as the first four reduced bases for $\log(\Lambda(M))$.

To obtain the coefficients $\boldsymbol{a} = \{a_k\}$ in Expansion (4) as functions of the parameters α, we construct implicit equations using an eigensystem of symmetric matrices (D. Figueroa et al. 2024, in preparation; P. Cook et al. 2024):

$$H(\alpha)\boldsymbol{a} = \lambda\boldsymbol{a}, \quad (5)$$

$$H(\alpha) = D_0 + \sum_i^m f_i(\alpha)H_i, \quad (6)$$

where the diagonal matrix D_0, as well as the symmetric matrix H_i, have yet undetermined coefficients that are eventually

Table 1
Parameters of the RBM Tidal Deformability Emulators for Each of the Three Parameters Set

Parameter Set	n	PMM dim	ρ (PC) (fm^{-3})	ρ (Norm) (fm^{-3})
Set 1	2	3	(0.60, 0.75)	(0.45, 0.60)
Set 2	6	2	[0.10–0.85]	[0.10–0.85]
Set 3	3	2	[0.10–0.85]	[0.10–0.85]

Note. The second column denotes the number of principal components n kept in Expansion (4), and the third column denotes the dimensions of the PMM used for emulating the norm of the coefficients $\boldsymbol{a}$. The fourth and fifth columns denote the density locations ρ where the EOS is evaluated to obtain the pressures that is the feature $f_i(\alpha)$ in Equation (6), with square brackets representing equidistant points within the specified range in steps of 0.05 fm^{-3}.

learned from data. The function f_i represents *features* of the controlling parameters α, such as polynomial expansions of α. In our case, we choose these features to be the pressure for the targeted EOS evaluated at various densities that are relevant for the NS structure. Since solving the system in Equation (5) gives an eigenvector $\boldsymbol{a}$ that is defined up to a normalization, we further emulate the normalization $N(\alpha)$ by directly using the parametric matrix model (PMM) described in P. Cook et al. (2024) with an expansion on analogous features $f_i(\alpha)$.[6] The details about the RBM data-driven emulator, including the dimensions for the normalization emulation using the PMM, are shown in Table 1.

To train the emulator and estimate the matrix elements of Equation (5), we use 2000 training samples for each EOS set. We then fit the matrix elements in Equation (6) for both the principal component (PC) and PMM emulators using the Levenberg–Marquardt least-squares routine (K. Levenberg 1944; D. W. Marquardt 1963), performing ten fits on the same training samples but with different initial conditions. Specifically for the PC emulator, we take the first normalized eigenvector corresponding to the largest eigenvalue of its H-matrix. For the PMM, we emulate the normalization by taking the largest eigenvalue of its H-matrix. We then validate each fit by calculating a χ^2 of the resultant emulator with respect to the samples not used in the fit and chose the emulator that minimizes this χ^2.

In Figure 5, we show the validation performance of the RBM for the five-parameter EOS model. The RBM can capture the general behavior of the full TOV solver, with the average of $\Delta_{1.4}$ being 2.4%. The largest outlier has a $\Delta_{1.4} \approx 60\%$, but only around five samples have such large uncertainties. Note that at higher NS masses, the RBM sometimes predicts an unphysical, nonmonotonous dependence of $\hat{\Lambda}$ on the NS mass since the RBM fails to emulate one of the PCs for the five-parameter model.

4. Comparison of the Emulators

In the following, we compare the performance of the three emulators in terms of both accuracy and computational speedup. We show the computational accuracy versus time plots in Figures 6–8 for the one-parameter, five-parameter and ten-parameter models, respectively. The validation uncertainty

is shown for stars with 1.4 $M_\odot$ (2 $M_\odot$) in the left (right) panels. For a fair comparison, all emulators are evaluated on a single CPU core for each validation sample.

From Figures 6–8, we observe that the MLP has the lowest average validation uncertainty among all emulators for all EOS models at both 1.4 $M_\odot$ and 2 $M_\odot$. This is followed by the MLP* and GP emulators, which have comparable uncertainties, with MLP* generally showing lower average uncertainties, except for the left panel in Figure 7. In all cases, the RBM has the largest validation uncertainties. Note that the comparison between MLP and MLP* allows us to gauge the impact of bagging; see Section 3.1. Additionally, we examined the performance of the MLP emulator with respect to the number of members in the ensemble included in the bagging algorithm. We found that the average validation uncertainty improves with the number of individual MLPs, before saturating at about 65 MLPs for the five and ten-parameter models. For the one-parameter model, bagging has a negligible impact; see Figure 6.

The RBM offers the largest computational speedup compared to the full TOV solver, which takes $\approx$3.5 s per sample. The MLP* emulator is similarly fast and a factor of 100 faster than MLP, as expected. The GP emulator is the slowest among the emulators but still achieves over an order of magnitude speedup compared to the full solver. We conclude that for applications requiring high accuracy, the MLP emulator is the best choice. However, if computational speedup is the primary concern, the MLP* or RBM emulators may be more practical, albeit with increased emulator uncertainties. Given the typical computational resources used for BNS parameter inference runs (G. Ashton et al. 2019; C. M. Biwer et al. 2019), emulators that can output the Λ–M sequence within $\approx$10^{-2} s are sufficiently fast (J. Veitch et al. 2015). The required emulator accuracy is harder to estimate since this depends on the signal-to-noise ratio of the data and the desired precision on the EOS parameters. However, for typical events that are expected to be observed in the upcoming years, we expect that emulators with $\Delta \approx 1\%$ would yield sufficiently accurate posterior distributions.

As mentioned earlier, the primary factor limiting both speed and accuracy of the GP is its inability to handle large data sets. Although we found that bagging improves the average uncertainty, the accuracy of the GP remains worse than both MLP and MLP*. A possible solution might be the implementation of a sparse GP (H. Liu et al. 2020), but this is beyond the scope of this work. In contrast, the MLP emulator scales remarkably well with the size of the training set, and we expect this improvement to continue for even larger training sets than those used in this work. Increasing the size of the training set is, therefore, a simple and effective way of improving the MLP and MLP* emulators in the future.

Finally, we note that the validation uncertainty is generally larger at 2 $M_\odot$ than 1.4 $M_\odot$. This is because the calculation of the percentage uncertainty requires dividing by the true tidal deformabilities, which are much smaller ($\approx$10) at 2 $M_\odot$ compared to their values at 1.4 $M_\odot$.

5. Example Analysis of a Potential Loud BNS Observation

To test our emulators in a Bayesian parameter inference, we now study a fictitious loud BNS observation that could be detected by next-generation GW detectors, such as Cosmic Explorer (M. Evans et al. 2021). We analyze the event using a simplified hierarchical approach (P. Landry & K. Chakravarti 2022) but will

---

[6] In our explorations, attempting to encode the normalization in the eigenvalue of the same equation system (Equation (5)) yielded worse results than training a separate PMM for the normalization alone.
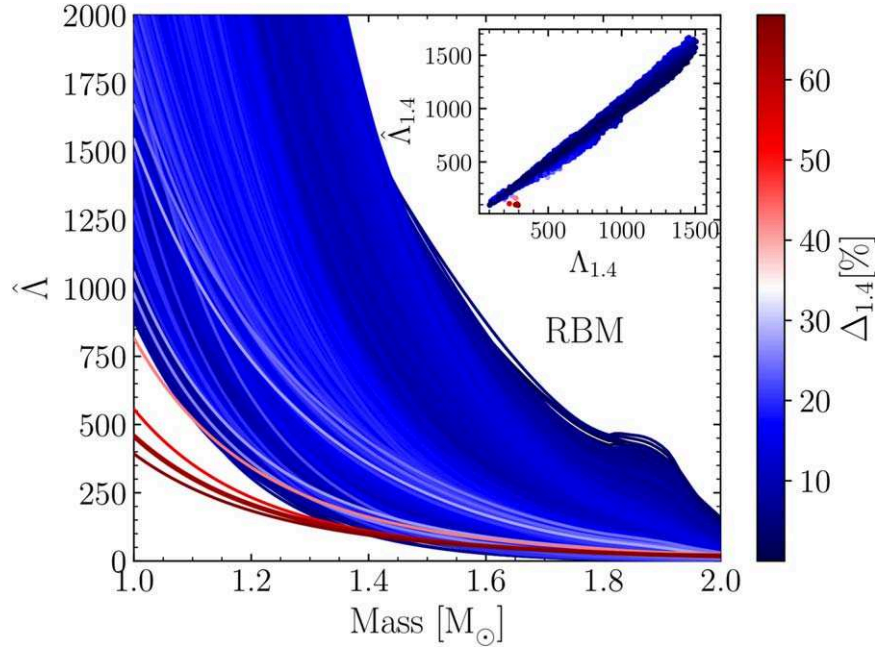
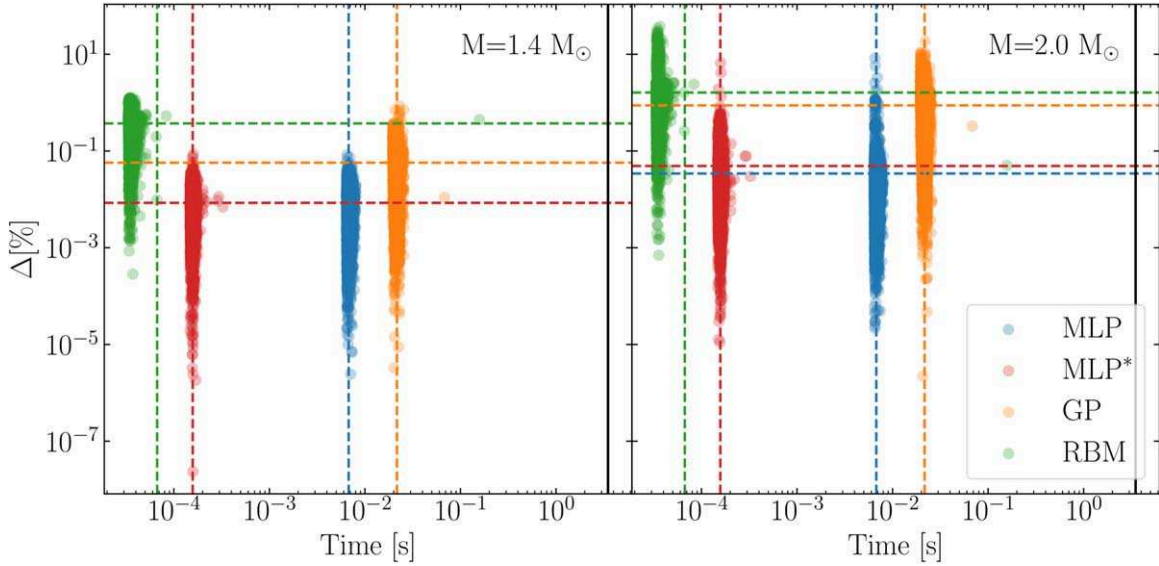**Figure 5.** Same as Figure 2 but for the RBM emulator.

Figure 6. Computational accuracy vs. time plot for the one-parameter model. This plot helps compare emulators' trade-offs in a similar way to the Pareto front (S. L. Brunton & J. N. Kutz 2019). Each dot corresponds to a validation sample, and we show a reduced set of only 5000 validation samples. The dashed lines indicate the averages over all 5000 validation samples. The total CPU time required to evaluate the full TOV solver is indicated by the solid black vertical line. Note that in the left panel, the lines for the MLP and MLP* overlap.

demonstrate our emulators in a full Bayesian parameter inference in future work. We use the analysis of this mock event to showcase the reliability of our emulators at reproducing the PDFs for EOS parameters. Here, we focus on the PDFs of two such parameters relevant for the nuclear symmetry energy to illustrate this point.

The posterior distribution for a set of EOS parameters α, given GW data d, can be written as

$$P(\alpha|d) \propto P(d|\alpha)P(\alpha)$$

$$\propto P(\alpha)\left[\int d\mathcal{M}_c dq d\tilde{\Lambda}\, P(d|\mathcal{M}_c, q, \tilde{\Lambda})P(\mathcal{M}_c, q, \tilde{\Lambda}|\alpha)\right],$$

$$(7)$$

where $\mathcal{M}_c$ is the chirp mass, q is the mass ratio, and $\tilde{\Lambda}$ is the combined dimensionless tidal deformability of the binary system (T. Hinderer et al. 2010). The function $P(\mathcal{M}_c, q, \tilde{\Lambda}|\alpha)$ is determined by a given EOS represented by α. Here, we fix the chirp mass of our injected event at 1.188 $M_\odot$, the central value observed in GW170817 (B. P. Abbott et al. 2017b) and allow for a uniform distribution of q between 0.6 and 1.0, $U(q)$. Therefore, the marginal likelihood of the BNS parameters $P(d|\mathcal{M}_c, q, \tilde{\Lambda}) \equiv \mathcal{L}(\mathcal{M}_c, q, \tilde{\Lambda})$ has the form

$$\mathcal{L}(\mathcal{M}_c, q, \tilde{\Lambda}) = \delta(\mathcal{M}_c - 1.188)U(q)\mathcal{L}(\tilde{\Lambda}). \qquad (8)$$
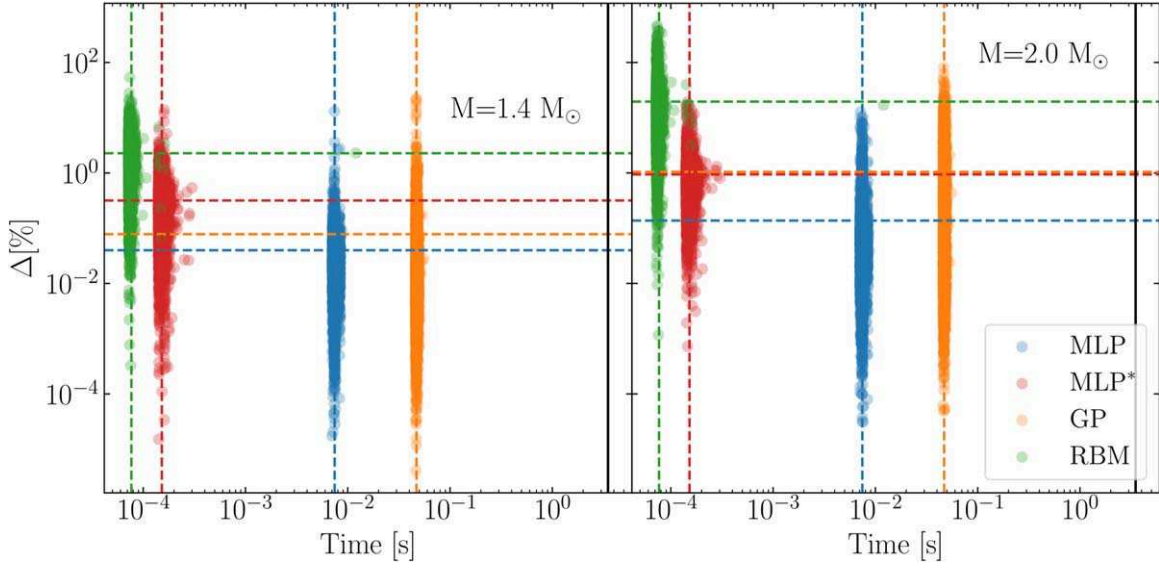
Figure 7. Same as Figure 6 but for the five-parameter model.
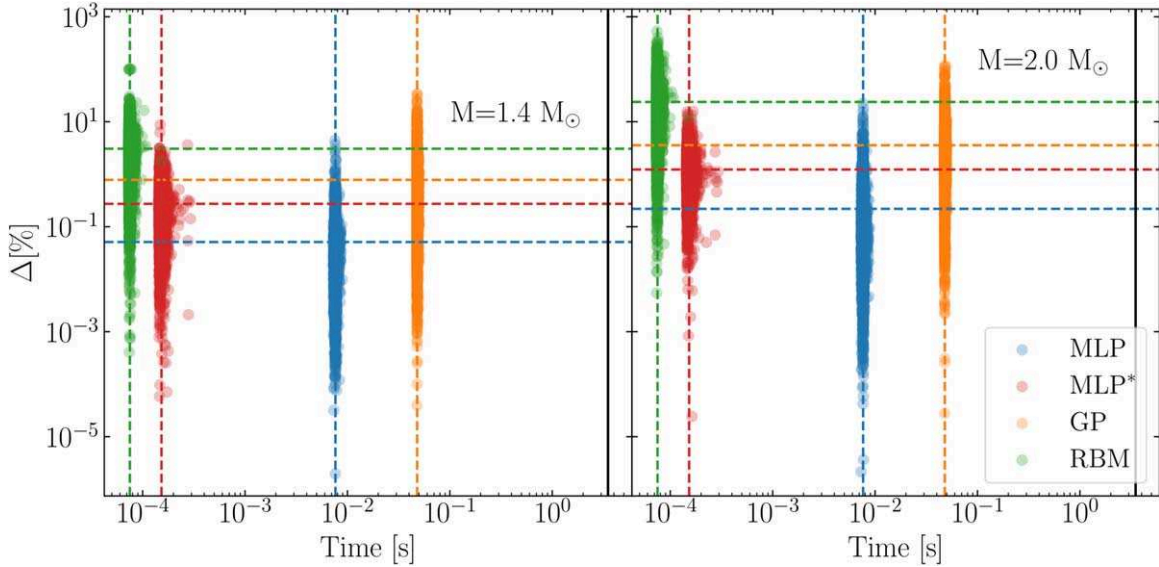

Figure 8. Same as Figure 6 but for the ten-parameter model.

For the factorized likelihood $\mathcal{L}(\tilde{\Lambda})$, we assume a simple Gaussian distribution centered at $\tilde{\Lambda} = 500$ with a standard deviation of 25, which corresponds to a 5% uncertainty. We have checked that the qualitative conclusions of this section do not change upon varying the mean and standard deviation of the Gaussian distribution. We then evaluate the posterior distribution on α using Equation (7). In practice, this is done by drawing a large number of samples from the prior $P(\alpha)$ and assigning each a likelihood weight proportional to the integral within the square brackets in Equation (7). For the integral over q, we use Simpson's method with a grid spacing of 0.01. Note that the remaining integral simplifies drastically under the above approximations.

For the samples drawn from the prior $P(\alpha)$, we employ our set of 100 K validation samples. Each validation sample

corresponds to the true Λ–M curve, calculated with the full TOV solver, as well as the predicted $\hat{\Lambda}$–M curve calculated using the emulators. Therefore, we obtain the posterior on α calculated from both the full solver as well as the emulators. Our results for the five-parameter model are shown in Figure 9, where we show the marginalized 1D posterior distribution on two NEPs, L_{sym} and K_{sym}. We find that the emulated PDFs for the GP and MLP match perfectly with the PDF of the full TOV solver. We have checked that this is the case for all five parameters of the EOS model. Results for the one-parameter and ten-parameter EOS models are qualitatively similar. While the RBM does not exactly reproduce the true PDF, the agreement is nevertheless very good. Overall, our results demonstrate the potential of our emulators for enhancing Bayesian parameter inference of BNS observations.
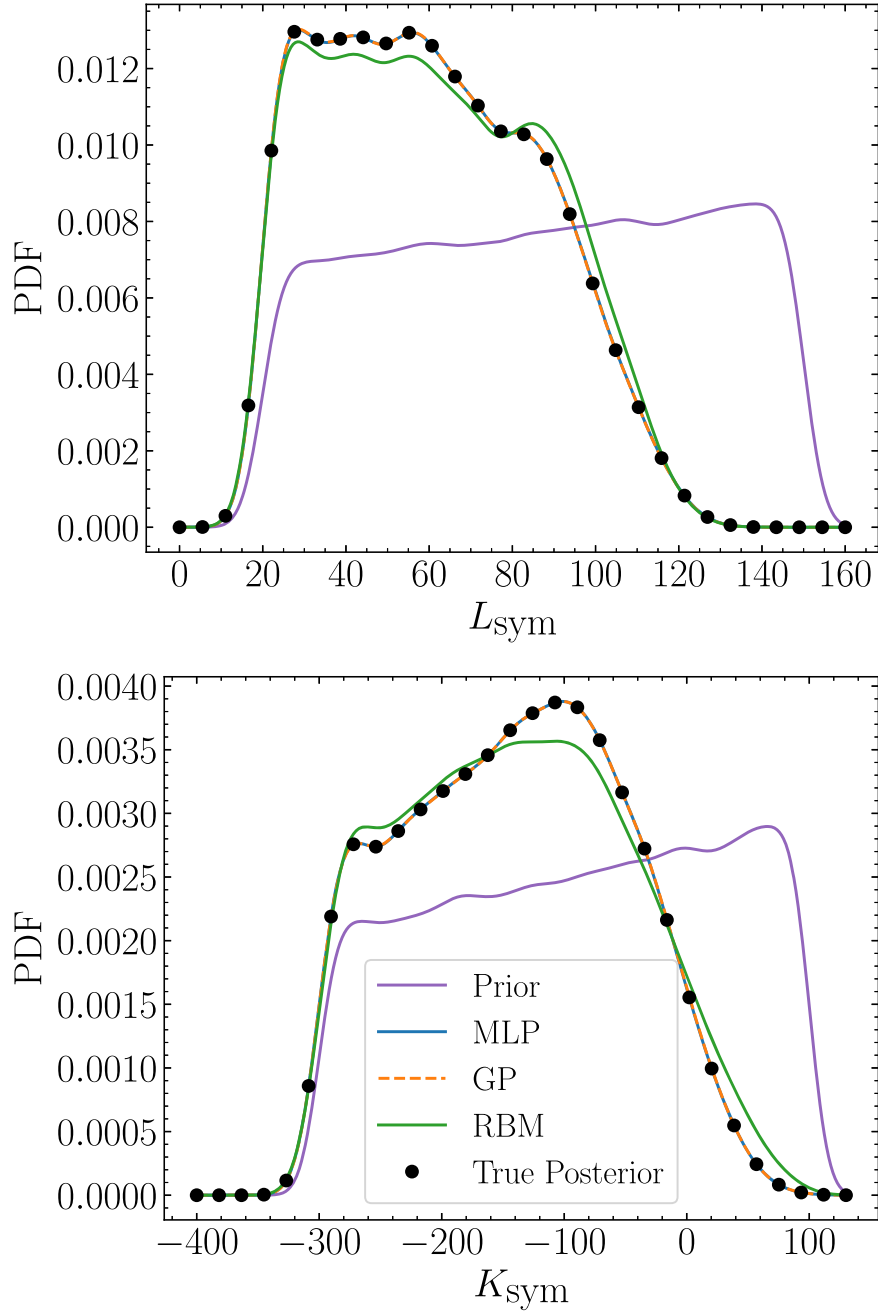
Figure 9. PDFs on two EOS model parameters, computed using the five-parameter model, for a fictitious high-precision measurement of the tidal deformability of a 1.4 $M_\odot$ NS described in the text. The PDFs for all emulators match well with that obtained using the full TOV solver (black circles).

6. Conclusion

To accelerate the inference of nuclear-physics parameters from astrophysical NS observations, we have developed three different algorithms for emulating solutions to the TOV equations. We have focused on emulating the tidal deformability as this is the key quantity for GW data analyses of BNS mergers. However, our framework can easily be extended to emulate other astrophysical NS observables such as the NS radii, moments of inertia, baryonic masses, Kepler masses, etc. We have developed our emulators for three different EOS models, two of which are capable of incorporating constraints from nuclear theory and experiment while simultaneously

being agnostic enough to allow for phase transitions at high densities. Additionally, the emulation procedures outlined here are applicable to many other EOS formalisms and are not exclusive to the metamodel with speed-of-sound extension. Our MLP emulator proves to be the most accurate, while the RBM offers the best speedup. All emulators are capable of predicting the full Λ–M sequence within 0.1 s. Finally, we have shown that the PDFs on EOS model parameters predicted by our emulators match very well with the true solution for a loud mock BNS signal with a 5% measurement uncertainty. Given these encouraging results, the work presented in this paper and further developments to our emulation strategies will pave the

way for future studies aiming to extract the nuclear EOS from observations of BNSs and other astrophysical sources including NSs.

ORCID iDs

Brendan T. Reed ⓘ https://orcid.org/0000-0002-7775-5423
Rahul Somasundaram ⓘ https://orcid.org/0000-0003-0427-3893
Soumi De ⓘ https://orcid.org/0000-0002-3316-5149
Cassandra L. Armstrong ⓘ https://orcid.org/0000-0002-1198-7774
Pablo Giuliani ⓘ https://orcid.org/0000-0002-8145-0745
Collin Capano ⓘ https://orcid.org/0000-0002-0355-5998
Duncan A. Brown ⓘ https://orcid.org/0000-0002-9180-5765
Ingo Tews ⓘ https://orcid.org/0000-0003-2656-6355

References

Abbott, B. P., Abbott, R., Abbott, T.D., et al. 2017a, ApJL, 848, L12
Abbott, B. P., Abbott, R., Abbott, T.D., et al. 2017b, PhRvL, 119, 161101
Abbott, B. P., Abbott, R., Abbott, T.D., et al. 2018, PhRvL, 121, 161101
Al-Mamun, M., Steiner, A. W., Nättilä, J., et al. 2021, PhRvL, 126, 061101
Annala, E., Gorda, T., Kurkela, A., Nättilä, J., & Vuorinen, A. 2020, NatPh, 16, 907
Annala, E., Gorda, T., Kurkela, A., & Vuorinen, A. 2018, PhRvL, 120, 172703
Antoniadis, J., Freire, P. C. C., Wex, N., et al. 2013, Sci, 340, 6131
Ashton, G., Hübner, M., Lasky, P.D., et al. 2019, ApJS, 241, 27
Bandopadhyay, A., Kacanja, K., Somasundaram, R., Nitz, A. H., & Brown, D. A. 2024, arXiv:2402.05056
Biwer, C. M., Capano, C. D., De, S., et al. 2019, PASP, 131, 024503
Bonilla, E., Giuliani, P., Godbey, K., & Lee, D. 2022, PhRvC, 106, 054322
Branchesi, M., Maggiore, M., Alonso, D., et al. 2023, JCAP, 07, 068
Breiman, L. 1996, Machine Learning, 24, 123
Breschi, M., Gamba, R., Carullo, G., et al. 2024, ApJ, 689, A51
Brunton, S. L., & Kutz, J. N. 2019, Data-driven Science and Engineering: Machine Learning, Dynamical Systems, and Control (Cambridge: Cambridge Univ. Press)
Burohman, A. M., Besselink, B., Scherpen, J. M., & Camlibel, M. K. 2023, ITAC, 68, 6160
Canizares, P., Field, S. E., Gair, J., et al. 2015, PhRvL, 114, 071104
Capano, C. D., Tews, I., Brown, S. M., et al. 2020, NatAs, 4, 625
Chen, Y., Gottlieb, S., Ji, L., & Maday, Y. 2021, JCoPh, 444, 110545
Cook, P., Jammooa, D., Hjorth-Jensen, M., Lee, D. D., & Lee, D. 2024, arXiv:2401.11694
Cromartie, H. T., Fonseca, E., Ransom, S. M., et al. 2019, NatAs, 4, 72
De, S., Finstad, D., Lattimer, J. M., et al. 2018, PhRvL, 121, 091102
Demorest, P., Pennucci, T., Ransom, S., Roberts, M., & Hessels, J. 2010, Natur, 467, 1081
Dietrich, T., Coughlin, M. W., Pang, P. T. H., et al. 2020, Sci, 370, 1450
Drischler, C., Hebeler, K., & Schwenk, A. 2019, PhRvL, 122, 042501
Drischler, C., Melendez, J., Furnstahl, R., Garcia, A., & Zhang, X. 2023, FrP, 10, 1092931
Essick, R., Landry, P., & Holz, D. E. 2020, PhRvD, 101, 063007
Evans, M., Adhikari, R. X., Afle, C., et al. 2021, arXiv:2109.09882
Field, S. E., Galley, C. R., Herrmann, F., et al. 2011, PhRvL, 106, 221102
Finstad, D., White, L. V., & Brown, D. A. 2023, ApJ, 955, 45
Flanagan, E. E., & Hinderer, T. 2008, PhRvD, 77, 021502
Fonseca, E., Cromartie, H. T., Pennucci, T. T., et al. 2021, ApJL, 915, L12
Giuliani, P., Godbey, K., Bonilla, E., Viens, F., & Piekarewicz, J. 2023, FrP, 10, 1054524
Greif, S. K., Raaijmakers, G., Hebeler, K., Schwenk, A., & Watts, A. L. 2019, MNRAS, 485, 5363
Güven, H., Bozkurt, K., Khan, E., & Margueron, J. 2020, PhRvC, 102, 015805
Han, M.-Z., Jiang, J.-L., Tang, S.-P., & Fan, Y.-Z. 2021, ApJ, 919, 11
Hebeler, K., Lattimer, J. M., Pethick, C. J., & Schwenk, A. 2013, ApJ, 773, 11
Hernandez Vivanco, F., Smith, R., Thrane, E., et al. 2019, PhRvD, 100, 103009
Hinderer, T. 2008, ApJ, 677, 1216
Hinderer, T., Lackey, B. D., Lang, R. N., & Read, J. S. 2010, PhRvD, 81, 123016
Huth, S., Pang, P. T. H, Tews, I., et al. 2022, Natur, 606, 276
Iacovelli, F., Mancarella, M., Mondal, C., et al. 2023, PhRvD, 108, 122006
Keller, J., Hebeler, K., & Schwenk, A. 2023, PhRvL, 130, 072701
Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980
Koehn, H., Rose, H., Pang, P. T. H, et al. 2024, arXiv:2402.04172
Komoltsev, O., Somasundaram, R., Gorda, T., et al. 2024, PhRvD, 109, 094030
Kunert, N., Pang, P. T. H., Tews, I., Coughlin, M. W., & Dietrich, T. 2022, PhRvD, 105, L061301
Landry, P., & Chakravarti, K. 2022, arXiv:2212.09733
Landry, P., & Essick, R. 2019, PhRvD, 99, 084049
Landry, P., Essick, R., & Chatziioannou, K. 2020, PhRvD, 101, 123007
Landry, P., & Poisson, E. 2014, PhRvD, 89, 124011
Lattimer, J. M., & Prakash, M. 2004, Sci, 304, 536
Levenberg, K. 1944, QApMa, 2, 164
Lindblom, L. 1992, ApJ, 398, 569
Lindblom, L. 2010, PhRvD, 82, 103011
Lindblom, L., & Indik, N. M. 2014, PhRvD, 89, 064003
Liodis, I., Smirniotis, E., & Stergioulas, N. 2024, PhRvD, 109, 104008
Liu, H., Ong, Y.-S., Shen, X., & Cai, J. 2020, IEEE Transactions on Neural Networks and Learning Systems, 31, 4405
Malik, T., Alam, N., Fortin, M., et al. 2018, PhRvC, 98, 035804
Margalit, B., & Metzger, B. D. 2019, ApJL, 880, L15
Margueron, J., Hoffmann Casali, R., & Gulminelli, F. 2018, PhRvC, 97, 025805
Marquardt, D. W. 1963, J. Soc. Ind. Appl. Math., 11, 431
McGinn, J., Mukherjee, A., Irwin, J., et al. 2024, arXiv:2403.17462
Miller, M. C., Lamb, F. K., Dittmann, A. J., et al. 2019, ApJL, 887, L24
Miller, M. C., Lamb, F. K., Dittmann, A. J., et al. 2021, ApJL, 918, L28
Nielsen, M. A. 2015, Neural Networks and Deep Learning (Determination Press), http://neuralnetworksanddeeplearning.com/
Odell, D., Giuliani, P., Beyer, K., et al. 2024, PhRvC, 109, 044612
Oppenheimer, J. R., & Volkoff, G. M. 1939, PhRv, 55, 374
Pang, P. T. H., Dietrich, T., Coughlin, M. W., et al. 2023, NatCo, 14, 8352
Petzold, L. 1983, SIAM Journal on Scientific and Statistical Computing, 4, 136
Phillips, D. R., Furnstahl, R. J., Heinz, U., et al. 2021, JPhG, 48, 072001
Postnikov, S., Prakash, M., & Lattimer, J. M. 2010, PhRvD, 82, 024016
Prakash, A., Gupta, I., Breschi, M., et al. 2024, PhRvD, 109, 103008
Punturo, M., Abernathy, M., Acernese, F., et al. 2010, CQGra, 27, 194002

Quarteroni, A., Manzoni, A., & Negri, F. 2015, Reduced Basis Methods for Partial Differential Equations: An Introduction, Vol. 92 (Berlin: Springer),

Quarteroni, A., & Rozza, G. 2014, Reduced Order Methods for Modeling and Computational Reduction, Vol. 9 (Berlin: Springer)

Raaijmakers, G., Riley, T. E., Watts, A. L., et al. 2019, ApJL, 887, L22

Rasmussen, C. E., & Williams, C. K. I. 2005, Gaussian Processes for Machine Learning (Cambridge, MA: MIT Press)

Read, J. S., Lackey, B. D., Owen, B. J., & Friedman, J. L. 2009, PhRvD, 79, 124032

Reed, B. T., Fattoyev, F. J., Horowitz, C. J., & Piekarewicz, J. 2021, PhRvL, 126, 172503

Reitze, D., Adhikari, R. X., Ballmer, S., et al. 2019, BAAS, 51, 35

Riley, T. E., Watts, A. L., Bogdanov, S., et al. 2019, ApJL, 887, L21

Riley, T. E., Watts, A. L., Ray, P. S., et al. 2021, ApJL, 918, L27

Rose, H., Kunert, N., Dietrich, T., et al. 2023, PhRvC, 108, 025811

Smith, R., Field, S. E., Blackburn, K., et al. 2016, PhRvD, 94, 044031

Soma, S., Wang, L., Shi, S., Stöcker, H., & Zhou, K. 2022, JCAP, 2022, 071

Somasundaram, R., Armstrong, C. L., Giuliani, P., et al. 2024, arXiv:2404.11566

Somasundaram, R., Drischler, C., Tews, I., & Margueron, J. 2021, PhRvC, 103, 045803

Somasundaram, R., Tews, I., & Margueron, J. 2023, PhRvC, 107, 025801

Tews, I., Carlson, J., Gandolfi, S., & Reddy, S. 2018a, ApJ, 860, 149

Tews, I., Margueron, J., & Reddy, S. 2018b, PhRvC, 98, 045804

Tiwari, P., & Pai, A. 2024, arXiv:2405.08163

Veitch, J., Raymond, V., Farr, B., et al. 2015, PhRvD, 91, 042003

Ventagli, G., & Saltas, I. D. 2024, arXiv:2405.17908

Walker, K., Smith, R., Thrane, E., & Reardon, D. J. 2024, PhRvD, 110, 043013

Wang, J. 2023, CSE, 25, 4