Published for SISSA by Springer

Received: *January 31, 2021*
Accepted: *February 21, 2021*
Published: *March 30, 2021*

Extensive studies of the neutron star equation of state from the deep learning inference with the observational data augmentation

Yuki Fujimoto,[a] **Kenji Fukushima**[a,b] **and Koichi Murase**[c]

[a] *Department of Physics, The University of Tokyo,*
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan
[b] *Institute for Physics of Intelligence (IPI), The University of Tokyo,*
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan
[c] *Center for High Energy Physics, Peking University,*
Beijing 100871, China
E-mail: fujimoto@nt.phys.s.u-tokyo.ac.jp, fuku@nt.phys.s.u-tokyo.ac.jp, murase@pku.edu.cn

Abstract: We discuss deep learning inference for the neutron star equation of state (EoS) using the real observational data of the mass and the radius. We make a quantitative comparison between the conventional polynomial regression and the neural network approach for the EoS parametrization. For our deep learning method to incorporate uncertainties in observation, we augment the training data with noise fluctuations corresponding to observational uncertainties. Deduced EoSs can accommodate a weak first-order phase transition, and we make a histogram for likely first-order regions. We also find that our observational data augmentation has a byproduct to tame the overfitting behavior. To check the performance improved by the data augmentation, we set up a toy model as the simplest inference problem to recover a double-peaked function and monitor the validation loss. We conclude that the data augmentation could be a useful technique to evade the overfitting without tuning the neural network architecture such as inserting the dropout.

Keywords: QCD Phenomenology

ArXiv ePrint: 2101.08156

https://doi.org/10.1007/JHEP03(2021)273

Contents

1 Introduction

In the central cores of neutron stars baryonic matter is highly compressed by the gravitational force. The inward force by gravity is balanced by the outward pressure which strongly depends on intrinsic properties of cold and dense matter subject to the strong interaction. The study of inner structures of the neutron star requires a relation between the pressure p and the energy density ε; namely, the equation of state (EoS), $p = p(\varepsilon)$, at $T = 0$ (see, e.g., refs. [1–5] for recent reviews). At the center the baryon number density n_B may reach 5–10 times as large as the saturation density of nuclear matter, i.e., $n_0 \simeq 0.16\,(\text{nucleons})/\text{fm}^3$.

Towards the model-independent EoS determination, we should solve quantum chromodynamics (QCD), which is the first-principles theory of the strong interaction. At the moment, however, the model-independent results are only available at limited ranges of densities. At low density range of $n_B \sim 1\text{–}2\,n_0$ we can apply *ab initio* methods combined with the nuclear force derived from Chiral Effective Theory (χEFT) with controlled uncertainty estimates [6–14] (see ref. [15] for a recent review). At asymptotically high densities of $n_B \gtrsim 50\,n_0$ perturbative QCD calculations reliably converge [16–21] (see ref. [22] for a recent review). The intermediate density region around $2\text{–}10\,n_0$, which is relevant for the neutron star structure, still lacks trustable QCD predictions (see also ref. [23] for a recent attempt to improve the convergence in the resummed perturbation theory). To tackle the intermediate density region from QCD, we need to develop non-perturbative methods such as the Monte-Carlo simulation of QCD on the lattice (lattice QCD), but the lattice-QCD application to finite density systems is terribly hindered by the notorious sign problem (see, e.g., ref. [24] for a review). This is why neutron-star-physics studies still rely on phenomenological EoS constructions: rather *ab initio* approaches [25, 26] near the saturation density, estimates employing the Skyrme interactions [27], the relativistic mean field theories [28], and the functional renormalization group [29], etc.

Meanwhile, we have witnessed significant advances in neutron star observations over the last decades. Now these advances have opened an alternative pathway for the model-independent extraction of the EoS by statistical methods. Observations include the Shapiro delay measurements of massive two-solar-mass pulsars [30–33], the radius determination of quiescent low-mass X-ray binaries and thermonuclear bursters [34–38] (see reviews [2, 39, 40] for discussions on the methods and associated uncertainties), detections of gravitational waves from binary mergers involving neutron stars by the LIGO-Virgo collaboration [41, 42] as well as the X-ray timing measurements of pulsars by the NICER mission [43, 44]. Typical observable quantities of neutron stars involve mass M, radius R, compactness M/R, tidal deformability Λ (and their variants, e.g., Love number k_2), quadrupole moment Q, moment of inertia I, etc. The gravitational waves from binary neutron star mergers provide us with the information about the tidal deformability. Also, the NICER mission particularly targets at the compactness M/R of stars by measuring the gravitational lensing of the thermal emission from the stellar surface. Some of these neutron star quantities are connected through the universal relations that are insensitive to the EoS details. Among such relations, the most well-known example is the I-Love-Q relation [45, 46], which relates the moment of inertia I, the Love number k_2, and the quadrupole moment Q (see refs. [4, 47] and reference therein for other universal relations and usages).

These observations have provided some insights on the EoS and invoked discussions about non-perturbative aspect of QCD: for example, the emergence of quark matter in the neutron star [48] and a certain realization of duality between the confined and the deconfined matter [49–57]. Moreover, these observations, particularly the gravitational wave observation, may provide a unique opportunity for studying the hadron-to-quark phase transition occurring in the binary star merger and even better sensitivity will be achieved with the future detector [58–60].

For the purpose of extracting the most likely EoS from the observational data, there are diverse statistical technologies. The commonly used method is the Bayesian inference [34–36, 61–66]. There are still other methods such as the one based on the Gaussian processes which is a variation of the Bayesian inferences with nonparametric representation of the EoS [67–69], etc. Despite numerous efforts to quantitatively constrain the EoS, it is still inconclusive what the genuine EoS should look like because of uncertainties from the assumed prior distribution in the Bayesian analysis. Therefore, the EoS inference program is in need of complementary tools. As an alternative to these conventional statistical methods, the *deep learning*; namely, the machine learning devising deep neural networks (NNs), has been successfully applied to a wide range of physics fields [70–76]. Several studies have already put the NN in motion in the context of gravitational wave data analysis of binary neutron star mergers [77–80]. Along these lines, in our previous publications [81, 82] we proposed a novel approach to the EoS extraction based on the deep machine learning (see also refs. [83–85] for related works).

In the present work we make further comprehensive analyses using the machine learning method developed previously [81, 82]. Here we give a brief synopsis of our machine learning approach to the neutron star EoS problem. Given an EoS, various pairs of stellar mass and radius, M-R, follow from the general relativistic structural equation, i.e., Tolman-Oppenheimer-Volkoff (TOV) equation [86, 87]. We will express the *inverse* operation of solving the TOV equation in terms of the deep NN, train the NN with sufficiently large data set of mock observations, and eventually obtain a regressor that properly infers the most likely EoS corresponding to the observational M-R inputs. While the other approaches to the neutron star EoS explicitly assume a prior distribution for the EoS probability distributions, our method only implicitly depends on the prior assumption. Therefore, in a sense that the implementations are based on different principles and the prior dependences appear differently, we can consider our method as complementary to other conventional approaches. We also note that independent algorithms would help us with gaining some information about systematics. For more detailed discussions on the prior, see section 2.2.

This paper is organized as follows. Section 2 provides a detailed account on our problem setting, i.e., mapping neutron star observables to the EoS. Then, we outline our method by setting forth our parametrization for the EoS and reviewing the machine learning technique in general. Section 3 is devoted for the methodology study using mock data. We describe our data generating procedure and training setups, and then give a brief overview of the evolution of learning processes, which motivates the later analyses in section 5. We showcase the typical examples out of the whole results, and then we carry out statistical analyses for the data as a whole. We confront our NN method with a different method; namely, the polynomial regression, and compare two methods to show that our method surpasses the polynomial regression. In section 4, we present the EoSs deduced from the several different neutron star observables in real. We explain our treatment of uncertainty quantification along with the possibility of a weak first-order phase transition in the deduced EoS results. In section 5 we study the efficiency of data augmentation in the context to remedy the overfitting problem. Finally, we summarize this work and discuss future outlooks in section 6. Throughout this paper we use the natural unit system with $G = c = 1$ unless otherwise specified.

2 Supervised learning for the EoS inference problem

In this section we explicitly define our problem and summarize the basic strategy of our approach with the supervised machine learning. We will explain the concrete setup in each subsection where we adjust the strategy in accordance with the goal of each subsection. In the present study we want to constrain the EoS from the stellar observables. The EoS and the observables are non-trivially linked by the TOV equation which gives a means to calculate the neutron star structure from the EoS input. Thus, constraining the EoS from the observables is the inverse process of solving the TOV equation, but this inverse problem encounters difficulties from the nature of the observations. In section 2.1 we formulate the inverse problem of the TOV equation and discuss its difficulties. We proceed to our approach to this problem using the supervised machine learning in section 2.2. We closely describe the EoS parametrization and the data generation in section 2.3. Then, we explain the design of the deep NN in section 2.4 and its training procedures in section 2.5.

2.1 TOV mapping between the EoS and the M-R relation

In the present analysis we focus on the mass M and the radius R as the neutron star observables. Given a boundary condition of the core pressure p_c, the observables M and R of such a neutron star can be determined by solving the TOV equation [86, 87]:

$$\frac{dp(r)}{dr} = -\frac{[\varepsilon(r) + p(r)][m(r) + 4\pi r^3 p(r)]}{r[r - 2m(r)]}\,, \tag{2.1}$$

$$m(r) = 4\pi \int_0^r r'^2 dr'\, \varepsilon(r')\,, \tag{2.2}$$

where r is the radial coordinate which represents a distance from the stellar center. The functions, $p(r)$ and $\varepsilon(r)$, are the pressure and the energy density (i.e., the mass density), respectively, at the position r. The function $m(r)$ represents a mass enclosed within the distance r. We can readily integrate these integro-differential equations from $r = 0$ with the initial condition, $p(r = 0) = p_c$, towards the outward direction. The radius of the neutron star, R, is defined by the surface condition: $p(R) = 0$. The mass of the neutron star is the total mass within R, i.e., $M = m(R)$.

The only missing information in the combined TOV equations (2.1) and (2.2) is a relationship between p and ε, which is nothing but the EoS, i.e., $p = p(\varepsilon)$. Once an EoS is fixed, we can draw a p_c-parametrized trajectory of $(M(p_c), R(p_c))$ in the M-R plane. This trajectory is called the M-R relation. Thus, the operation of solving the TOV equation can be considered as a mapping from the functional space of the EoSs onto the functional space of the M-R relations. Here we call this mapping the “TOV mapping” denoted by Ψ_{TOV}:

$$\Psi_{\mathrm{TOV}} : \quad (\mathrm{EoS}) \quad \mapsto \quad (M\text{-}R \text{ relation})\,. \tag{2.3}$$

It is known that the mapping is invertible, i.e., the inverse mapping Ψ_{TOV}^{-1} exists [88]. Thus, *ideally*, one could uniquely reconstruct the EoS from an observed M-R relation. However, *practically*, we cannot access the entire continuous curve on the M-R relation (which we will briefly call the M-R curve) from neutron star observations. In reconstructing the EoS from the observational data, we are facing twofold obstacles:

1. *Incompleteness* — M-R data from the observation does not form a continuous curve. Since one neutron star corresponds to one point on the M-R curve, the observation of neutron stars gives a limited number of M-R points and only partial information on the original M-R curve is available. Even with a sufficiently large number of neutron stars, the information on possible unstable branches of the M-R curve, on which the neutron stars do not exist, would be missing.

2. *Uncertainties* — Real data is not a point but a distribution. Each M-R point is accompanied by observational uncertainties represented as a credibility distribution. Moreover, the central peak position of the distribution is not necessarily on the genuine M-R relation.

We can introduce the following symbolic representation of the above complication associated with the observational limitation:

$$\Pi_{\mathrm{obs}}(\omega): \quad (M\text{-}R \text{ relation}) \quad \mapsto \quad (M\text{-}R \text{ data})_{\omega}\,. \tag{2.4}$$

This mapping by $\Pi_{\mathrm{obs}}(\omega)$ is a convolution of a projection from the M-R relation to a set of points along the M-R curve and a probability distribution due to observational uncertainties. The distribution generally depends on observational scenarios (how many and what kinds of neutron stars are detected by which signals) and such dependence is collectively represented by ω. With this mapping, the relation between the EoS and the observed M-R data is more correctly expressed as

$$\Pi_{\mathrm{obs}}(\omega) \circ \Psi_{\mathrm{TOV}}: \quad (\mathrm{EoS}) \quad \mapsto \quad (M\text{-}R \text{ data})_{\omega}\,. \tag{2.5}$$

Because of this additional filter by $\Pi_{\mathrm{obs}}(\omega)$, one cannot reconstruct the correct EoS from the observational data just by calculating Ψ_{TOV}^{-1}. Since a part of original information is lost through $\Pi_{\mathrm{obs}}(\omega)$, this is an ill-posed problem, i.e., there is no unique well-defined answer due to insufficient information and uncertainties inherent in this problem.

Here, instead of reconstructing the *unique EoS*, we try to infer the *most likely EoS* from M-R observations with uncertainty; that is, the regression analysis of the inverse mapping including $\Pi_{\mathrm{obs}}(\omega)$ (see figure 1 for a schematic illustration). In particular it is crucial to develop an approach leading to outputs robust against the observational uncertainties of the M-R data.

2.2 General strategy for the machine learning implementation

To solve this ill-posed problem we employ a method of machine learning with deep NNs to infer the neutron star EoS. In particular we consider the supervised machine learning

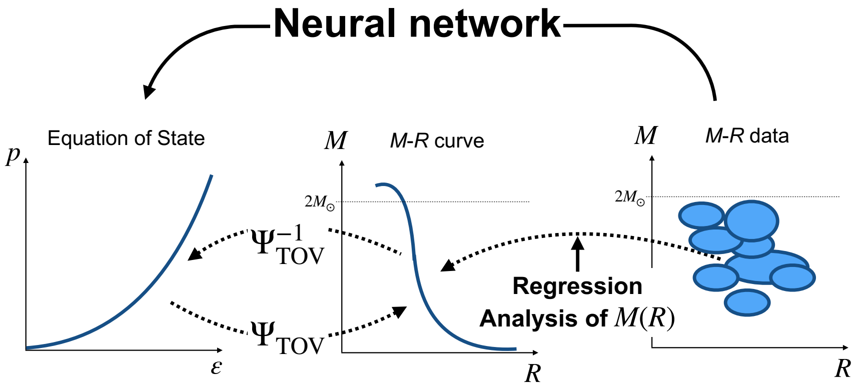


Figure 1. Schematic of the TOV mapping Ψ_{TOV} and the regression analysis of the inverse TOV mapping from the M-R data.

to train the NN with as many and different ω's as possible. The trained NN can then receive the M-R data for one particular ω and return the output of the most likely EoS correspondingly. To put it another way, the "inversion" of the mapping $\Pi_{\mathrm{obs}} \circ \Psi_{\mathrm{TOV}}$ is approximated by the regression, particularly by the deep NN in our approach, which would be symbolically written as $\mathrm{Reg}[\Pi_{\mathrm{obs}} \circ \Psi_{\mathrm{TOV}}]^{-1}$.

In the supervised learning, we need to prepare the training data and the validation data composed of pairs of the input M-R data and the desired output EoS. To generate the training data efficiently, we make use of the asymmetry between $\Pi_{\mathrm{obs}} \circ \Psi_{\mathrm{TOV}}$ and $\mathrm{Reg}[\Pi_{\mathrm{obs}} \circ \Psi_{\mathrm{TOV}}]^{-1}$; that is, we can straightforwardly calculate the forward mapping of $\Pi_{\mathrm{obs}} \circ \Psi_{\mathrm{TOV}}$ by modeling the observation $\Pi_{\mathrm{obs}}(\omega)$, while the latter inverse mapping, which is what we currently want to know, is more non-trivial. Thus, first, we randomly generate many possible answer EoSs represented by several parameters. We next generate the corresponding M-R data by applying the forward mapping with various ω's. We will explain technical details of handling and simplifying ω in section 2.3.

Now, we turn to the description of the architecture of our NN model and we will optimize the model parameters so that the model can infer the answer EoS corresponding to the input training M-R data. During the training, it is important to monitor the training quality by checking the prediction error behavior for the validation data. After the optimization process is complete with good convergence of the validation error, we can test the predictive power using the mock data for which the true answer EoS is known (see section 3 for actual calculations). Once the method passes all these qualification tests, we finally proceed to the application of the real observational data to obtain the most realistic EoS (see section 4 for details).

Here, we comment on an alternative possibility of the inference problem formulation. As already mentioned, the inverse TOV mapping, Ψ_{TOV}^{-1}, is well-defined by itself. Thus, it is also feasible to decompose the inference as $\mathrm{Reg}[\Pi_{\mathrm{obs}} \circ \Psi_{\mathrm{TOV}}]^{-1} = \Psi_{\mathrm{TOV}}^{-1} \circ \mathrm{Reg}[\Pi_{\mathrm{obs}}]^{-1}$ and train the NN aimed to approximate $\mathrm{Reg}[\Pi_{\mathrm{obs}}]^{-1}$; that is, the NN model would predict the M-R curve from the input M-R data. We will not adopt this strategy since it is unclear in this approach how to measure the reconstruction performance of the M-R curve. Our target space is the EoS space, but the distance in the M-R space is not uniformly reflected in the distance in the desired EoS space. We are ultimately interested in the EoS, so it is natural to directly model $\mathrm{Reg}[\Pi_{\mathrm{obs}} \circ \Psi_{\mathrm{TOV}}]^{-1}$ by maximizing the performance for the EoS reconstruction.

We also make a remark on the *discriminative modeling* and the *generative modeling* to clarify a difference between our machine learning method and other statistical approaches such as the Bayesian inference. The approach we are advocating in this paper belongs to the discriminative modeling: our NN model directly predicts the response variable, i.e., the EoS, for a fixed set of explanatory variables, i.e., the M-R data. In contrast, the statistical approaches such as the Bayesian inference for the neutron star EoS are to be categorized as the generative modeling. The Bayesian inference first models occurrence of the combination of the explanatory and response variables by the joint probability, $\Pr(M\text{-}R\ \mathrm{data}, \mathrm{EoS}) \equiv \Pr(M\text{-}R\ \mathrm{data}|\mathrm{EoS})\Pr(\mathrm{EoS})$. Here, $\Pr(\mathrm{EoS})$ models the prior belief on the EoS, and $\Pr(M\text{-}R\ \mathrm{data}|\mathrm{EoS})$ models the observation process. Then, the Bayes theorem gives a posterior distribution of EoS, $\Pr(\mathrm{EoS}|M\text{-}R\ \mathrm{data})$, from the joint probability, which corresponds to the likely EoS distribution for given M-R data.

These discriminative and generative approaches can be considered to be complementary to each other. They have the pros and cons as follows:

- The advantage in the generative approaches is that the EoS prediction for a given M-R data takes a distribution form, so that the prediction uncertainties can be estimated naturally. In the discriminative model, on the other hand, additional analysis is needed to estimate the prediction uncertainties. We will perform the additional uncertainty analysis within our method in section 4.3. At the same time, we will utilize our uncertainty analysis for a physics implication as discussed in section 4.5.

- In the discriminative modeling, once the NN training is completed, the most likely EoS corresponding to a given M-R data can immediately result from the mapping represented by the NN: the NN models the inference procedure by itself. Thus, the computational cost is much smaller than statistical analyses, which enables us to perform supplementary estimation on observables and/or quality assessments if necessary. In the generative approaches, on the other hand, the posterior calculation is computationally expensive because the EoS parameter space is large. In addition, one needs to calculate the posterior for each M-R data separately. If one applies the Bayesian inference to the real observational data only once, this computational cost would not matter. The computational cost becomes problematic, however, if one wants to examine the statistical nature of the inference using a large number of generated EoSs and M-R data.

2.3 Random EoS generation with parametrization by the speed of sound

The training data consists of pairs of the input M-R data and the output EoS, $p(\varepsilon)$, parametrized by a finite number of variables. We first generate EoSs by randomly assigning values to the EoS parameters, solve the TOV equation to find the corresponding M-R

curve, and finally generate the M-R data. The M-R data generation involves $\Pi_{\rm obs}(\omega)$, namely, we sample observational points of neutron stars on the M-R curve, and probabilistically incorporate finite observational uncertainties, which should mimic observations in the scenario ω. The real M-R data is composed of the probability distribution for each neutron star on the M-R plane, but in practice we should simplify ω and need to parametrize the M-R data as well. We will discuss the parametrization later and in this section we focus on our EoS parametrization and generation.

In designing the EoS parametrization, it is important to consider its affinity to the physical constraints such as the thermodynamic stability and the causality, i.e, p should be a monotonically non-decreasing function of ε, and the speed of sound, c_s, should not be larger than the speed of the light. For this reason, c_s, which is derived from

$$c_s^2 = \frac{\partial p}{\partial \varepsilon}\,, \tag{2.6}$$

is the convenient choice as the EoS parameters to impose the physical constraints easily.

Also, we need to smoothly connect the parametrized EoS to the empirical EoS known from the low density region. Up to the energy density of the saturated nuclear matter, ε_0, we use a conventional nuclear EoS, specifically SLy4 [27]. Because the EoS is well constrained by the saturation properties and the symmetry energy measurements near the saturation density, this specific choice of SLy4 would not affect our final results. In the energy region above ε_0 we employ the standard piecewise polytropic parametrization and partition a density range into a certain number of segments. In our present analysis we choose the density range, $[\varepsilon_0,\ 8\varepsilon_0]$, and six energy points, ε_i $(i = 0, \ldots, 5)$ with $\varepsilon_5 = 8\varepsilon_0$, equally separated in the logarithmic scale. Then, $(\varepsilon_{i-1},\ \varepsilon_i)$ $(i = 1, \ldots 5)$ form 5 segments. To be more specific, these values are $(\varepsilon_0, \varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5) = (150, 227, 345, 522, 792, 1200)\,\text{MeV/fm}^3$. The EoS parameter is the average speed of sound, $c_{s,i}^2 \equiv \langle c_s^2 \rangle = \langle \partial p/\partial\varepsilon \rangle$ $(i = 1, \ldots, 5)$, in i-th segment. From these definitions we see that the pressure values at the i-th segment boundaries, p_i, are read as[1]

$$p_i = p_{i-1} + c_{s,i}^2(\varepsilon_i - \varepsilon_{i-1})\,, \tag{2.7}$$

where p_0 is determined by $p_0 = p(\varepsilon_0)$ from our chosen nuclear EoS, i.e., SLy4. We make polytropic interpolation for the EoS by $p = k_i\varepsilon^{\gamma_i}$ whose exponent and coefficient are given by

$$\gamma_i = \frac{\ln(p_i/p_{i-1})}{\ln(\varepsilon_i/\varepsilon_{i-1})}\,, \qquad k_i = \frac{p_i}{\varepsilon_i^{\gamma_i}}\,. \tag{2.8}$$

[1] We can confirm that $c_{s,i}^2$ is indeed an average in each segment under this construction as

$$\begin{aligned}\langle c_s^2\rangle &\equiv \int_{\varepsilon_{i-1}}^{\varepsilon_i} \frac{d\varepsilon}{\varepsilon_i - \varepsilon_{i-1}} c_s^2 = \int_{\varepsilon_{i-1}}^{\varepsilon_i} \frac{d\varepsilon}{\varepsilon_i - \varepsilon_{i-1}} \frac{\partial p}{\partial \varepsilon} \\ &= \frac{1}{\varepsilon_i - \varepsilon_{i-1}} \int_{p_{i-1}}^{p_i} dp = \frac{p_i - p_{i-1}}{\varepsilon_i - \varepsilon_{i-1}} = c_{s,i}^2.\end{aligned}$$

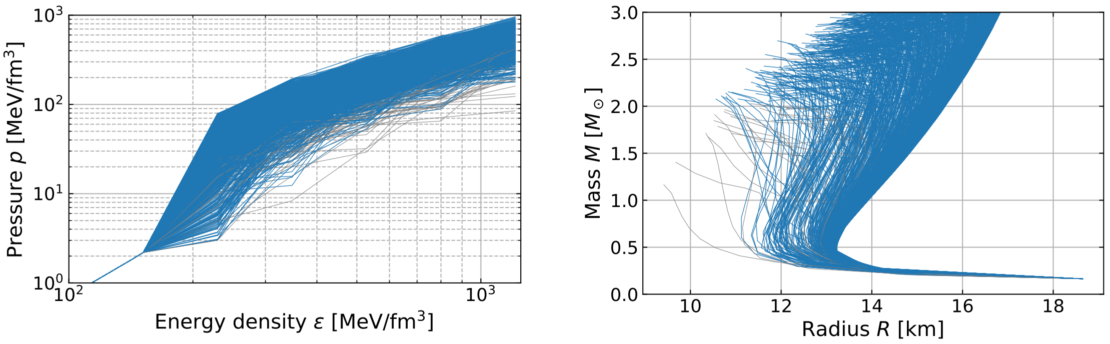


Figure 2. Randomly generated EoSs (left) and the corresponding M-R curves (right).

For the EoS generation we randomly assign the average sound velocities, $c_{s,i}^2$, of the uniform distribution within $\delta \leq c_{s,i}^2 < 1 - \delta$. The upper bound comes from the causality $c_s^2 < c^2 = 1$, and the lower bound from the thermodynamic stability $\partial p / \partial \varepsilon \geq 0$. Here, a small margin by $\delta = 0.01$ is inserted as a regulator to avoid singular behavior of the TOV equation. We here note that we allow for small c_s^2 corresponding to a (nearly) first-order phase transition. Repeating this procedure, we generate a large number of EoSs. For each generated EoS, we solve the TOV equation to obtain the M-R curve as explained in section 2.1. In figure 2 we show the typical EoS data and the corresponding M-R curves used in our later analyses. It should be noted here that we excluded artificial biases in the EoS generation and allowed for any possible EoSs including ones disfavored by the model calculations or the observational data (e.g., an EoS whose maximum mass exceeds $3M_\odot$ in the right panel of figure 2). This is important for the NN to be free from biases coming from specific model preference. In this way our analysis gains an enough generalization ability, covers even exotic scenario cases, and captures the correct underlying relations between the input and the output.

Before closing this part, let us mention a possible improvement for the random EoS generation, though we would not utilize this improvement to keep our present approach as simple as possible. The problem arises from our assumed uniform distribution of $c_{s,i}^2$. The parametrization and generation algorithm as explained above definitely allows for a first-order phase transition for sufficiently small $c_{s,i}^2$'s; however, due to the uniform distribution, it is a tiny percentage among the whole data for the generated EoSs to accommodate a first-order phase transition, and this may be a part of our "prior" dependence. Since a strong first-order transition scenario has already been ruled out from phenomenology, this prior dependence should be harmless. Nevertheless, if necessary, we can naturally increase the percentage of the EoSs with a first-order phase transition in a very simple way as follows: In eq. (2.8) we carry out the linear interpolation in the log-log plane. Alternatively, we can use the *spline* interpolation in the log-log plane. With such a smooth interpolation, there appear the energy density regions with negative $\partial p / \partial \varepsilon$, i.e., the EoS can be non-monotonic. We can then replace this non-monotonic part by the first-order phase transition using the Maxwell construction. This is one effective and natural way to enhance a finite fraction of

EoSs with a first-order phase transition while keeping the same $c_{s,i}^2$'s. We also note that one more merit in this procedure is that the end points of the first-order energy regions are not restricted to be on the segment boundaries. Thus, the strength of the first-order phase transition is not necessarily discretized by the segment width but an arbitrarily weak first-order phase transition could be realized with the same parametrization. We say again that we would not adopt this procedure: the present observational precision gives a tighter limitation and for the moment we do not benefit from this EoS improvement. In the future when the quality and the quantity of the observational data ameliorate, the above mentioned procedure could be useful.

2.4 Neural network design: a general introduction

The neural network, NN, is one representation of a function with fitting parameters. The deep learning, i.e., the machine learning method using a deep NN, is nothing but the optimization procedure of the parameters contained in the function represented by the NN. In particular, the supervised learning can be considered as the fitting problem (namely, *regression*) for given pairs of the input and the output, i.e., the training data. We often refer to a NN "model" to mean the optimized system of the function given in terms of the NN.

The advantage of the deep learning, as compared to naive fitting methods, lies in the generalization properties of the NN. We need not rely on any preknowledge about reasonable forms of fitting functions: the NN with a sufficient number of neurons is capable of capturing any continuous functions [89, 90]. With a large number of neurons (and layers), there are many fitting parameters, and recent studies of machine learning have developed powerful optimization methods for such complicated NNs.

The model function of feedforward NN can be expressed as follows:

$$\boldsymbol{y} = f(\boldsymbol{x}|\{W^{(1)}, \boldsymbol{b}^{(1)}, \dots, W^{(L)}, \boldsymbol{b}^{(L)}\}) \,, \tag{2.9}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are input and output, respectively. The fitting parameters, $\{W^{(\ell)}, \boldsymbol{b}^{(\ell)}\}_{\ell=1}^{L}$, are given in the form of matrices and vectors, respectively. For calculation steps, we first prepare $(L+1)$ *layers* (including the input and the output layers). Each layer $\boldsymbol{x}^{(\ell)}$ $(\ell = 0, \dots L)$ consists of nodes called *neurons* corresponding to the vector components, $x_1^{(\ell)}, \dots, x_{n_\ell}^{(\ell)}$. We note that the number of neurons, n_ℓ, can be different for different ℓ. The input $\boldsymbol{x}$ is set to the first layer $\boldsymbol{x}^{(0)}$ (which is labeled as $\ell = 0$ in this paper), and the subsequent layers are calculated iteratively as

$$\boldsymbol{x}^{(0)} = \boldsymbol{x} \,, \tag{2.10}$$

$$\boldsymbol{x}^{(\ell)} = \sigma^{(\ell)}(W^{(\ell)}\boldsymbol{x}^{(\ell-1)} + \boldsymbol{b}^{(\ell)}) \quad (\ell = 1, \dots, L) \,. \tag{2.11}$$

The L-th layer becomes the final output, $\boldsymbol{y} = f(\boldsymbol{x}) = \boldsymbol{x}^{(L)}$. Here, $\sigma^{(\ell)}(x)$'s are called *activation functions*, whose typical choices include the sigmoid function: $\sigma(x) = 1/(e^x + 1)$, the ReLU: $\sigma(x) = \max\{0, x\}$, hyperbolic tangent: $\sigma(x) = \tanh(x)$, etc. In the above notation $\sigma^{(\ell)}$ returns a vector by applying the activation to each vector component of the function argument, i.e., $\sigma(\boldsymbol{x}) = \big(\sigma(x_1), \dots, \sigma(x_n)\big)^{\mathrm{T}}$. The fitting parameters, $W^{(\ell)}$ and $\boldsymbol{b}^{(\ell)}$, on the ℓ-th layer, denote the *weights* between nodes in the two adjacent layers and

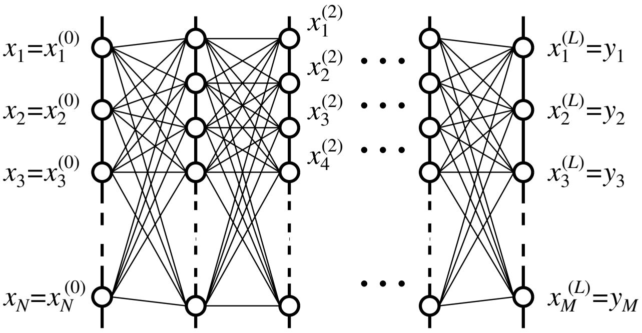


Figure 3. Schematic illustration of the feedforward neural network.

the activation offset at each neuron called *bias*, respectively. These fitting parameters have no physical interpretation at the setup stage. The general architecture is schematically depicted in figure 3, in which the calculation proceeds from the left with input $\boldsymbol{x}$ to the right with output $\boldsymbol{y}$.

The complexity choice of the NN structure, such as the number of layers and neurons, has a tradeoff problem between the fitting power and the overfitting. To capture the essence of the problem, the complexity of layers and neurons should be sufficiently large. At the same time, to avoid the overfitting problem, and to train the NN within reasonable time, the number of layers and neurons should not be too large. There is no universal prescription and the performance depends on the problem to be solved. For our setup we found overall good performance when we chose the neuron number on the first layer greater than the input neuron number. A more sophisticated NN may be possible, but for our present purpose a simple network structure as in figure 3 is sufficient.

2.5 Loss function and training the neural network

For the actual procedure of machine learning we need to choose a *loss function* and minimize it, which is called *training*. We define a loss function as

$$\mathcal{L}(\{W^{(\ell)},\boldsymbol{b}^{(\ell)}\}_\ell) \equiv \int d\boldsymbol{x}\,\mathrm{Pr}(\boldsymbol{x})\ell\Big(\boldsymbol{y}, f(\boldsymbol{x}|\{W^{(\ell)},\boldsymbol{b}^{(\ell)}\}_\ell)\Big)\,, \tag{2.12}$$

where $\ell(\boldsymbol{y},\boldsymbol{y}')$ quantifies the distance or error between the NN prediction $\boldsymbol{y}'$ and the answer $\boldsymbol{y}$ provided in the training data. The exact formula for the distribution $\mathrm{Pr}(\boldsymbol{x})$ is not necessarily known in general. Moreover, even if it is known, multidimensional integration in eq. (2.12) is practically intractable, so let us approximate it with the sum over the samples in the training data, $\mathcal{D}=\{(\boldsymbol{x}_n,\boldsymbol{y}_n)\}_{n=1}^{|\mathcal{D}|}$, with $|\mathcal{D}|$ denoting the sample size:

$$\mathcal{L}(\{W^{(\ell)},\boldsymbol{b}^{(\ell)}\}_\ell) \approx \frac{1}{|\mathcal{D}|}\sum_{n=1}^{|\mathcal{D}|}\ell\Big(\boldsymbol{y}_n, f(\boldsymbol{x}_n|\{W^{(\ell)},\boldsymbol{b}^{(\ell)}\}_\ell)\Big)\,. \tag{2.13}$$

The optimization problem we deal with here is to minimize the above loss function by tuning the network parameters $\{W^{(\ell)},\boldsymbol{b}^{(\ell)}\}_\ell$. The minimization is usually achieved with the stochastic gradient descent method or its improved versions. The deterministic methods would be easily trapped in local minima or saddle points of the loss function. In practice, therefore, the stochastic algorithms are indispensable. In gradient descent method the parameters $\{W^{(\ell)}(t), \boldsymbol{b}^{(\ell)}(t)\}_\ell$ at the iteration step t are updated along the direction of the derivative of the loss function with respect to the parameters:

$$W^{(\ell)}(t+1) = W^{(\ell)}(t) - \eta \frac{\partial \mathcal{L}(W^{(\ell)}(t))}{\partial W^{(\ell)}(t)} \,, \tag{2.14}$$

where η is called learning rate. We can update $\boldsymbol{b}^{(\ell)}$ in the same way. We usually evaluate the derivatives by using the *mini-batch method*. In the mini-batch method, the training data set is first randomly divided into multiple subsets, which is called *mini-batches*. Then, the derivative of the loss function is estimated within a mini-batch $\mathcal{B}$. Using the explicit expression (2.13), the approximated derivatives read:

$$\frac{\partial \mathcal{L}(W^{(\ell)})}{\partial W^{(\ell)}} \approx \frac{1}{|\mathcal{B}|} \sum_{n=1}^{|\mathcal{B}|} \frac{\partial \ell\Big(\boldsymbol{y}_n, f(\boldsymbol{x}_n | \{W^{(\ell)}, \boldsymbol{b}^{(\ell)}\}_\ell)\Big)}{\partial W^{(\ell)}} \,, \tag{2.15}$$

where the *batch size* $|\mathcal{B}|$ denotes the number of sample points in $\mathcal{B}$, whose optimal choice differs case-by-case. The *epoch* counts the number of scans over the entire training data set $\mathcal{D}$. The parameters are updated for each mini-batch, so one epoch amounts to $|\mathcal{D}|/|\mathcal{B}|$ updates. Usually, the derivative, $\partial\ell/\partial W$ appearing in eq. (2.15), is evaluated by a method called backpropagation.

For numerics we basically use a Python library, Keras [91] with TensorFlow [92] as a backend. In this work we specifically employ `msle`, i.e., the mean square logarithmic errors, for the choice of the loss $\ell(\boldsymbol{y}, \boldsymbol{y}')$ appearing in eq. (2.12); that is,

$$\ell_{\texttt{msle}}(\boldsymbol{y}, \boldsymbol{y}') \equiv \left|\log \boldsymbol{y} - \log \boldsymbol{y}'\right|^2 \,. \tag{2.16}$$

We use a specific fitting algorithm, Adam [93], with the mini-batch size being 100 for the analysis in section 3 and 1000 for the analysis in section 4. We initialize our NN parameters with the Glorot uniform distribution.

3 Extensive methodology study and the performance tests

This section is an extensive continuation from our previous work on the methodology study [81]. We here present more detailed and complete investigations of various performance tests.

In this section, to put our considerations under theoretical control, we assume ω to be a hypothetical observational scenario that 15 neutron stars are observed and $(M_i,\, R_i)$ $(i = 1, \ldots, 15)$ are known with a fixed size of observational uncertainty following the previous setup in refs. [2, 37, 38, 81]. We design a NN, generate the validation data as well as the training data, and train the NN to predict the EoS parameters, $c_{s,i}^2$, from the artificially generated M-R data, $(M_i,\, R_i)$ $(i = 1, \ldots, 15)$ (i.e., mock data). We discuss the training behavior of the NN, the reconstruction performance of the EoS and the M-R curve, and the distribution of reconstruction errors.

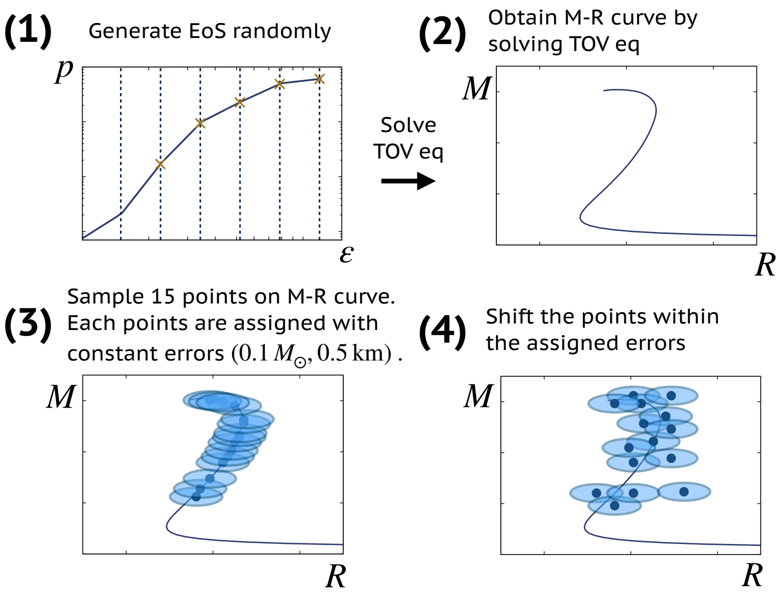

(1)
Generate EoS randomly
p
ε
Solve TOV eq
(2)
Obtain M-R curve by solving TOV eq
M
R
(3)
Sample 15 points on M-R curve. Each points are assigned with constant errors $(0.1\,M_{\odot}, 0.5\,\mathrm{km})$.
M
R
(4)
Shift the points within the assigned errors
M
R

Figure 4. Schematic flow of data generation procedures for the analysis in section 3.

3.1 Mock data generation and training

For the current study in this section, we deal with the input M-R data, (M_i, R_i) $(i = 1, \ldots 15)$, and the output EoS parameters, $c^2_{s,i}$ $(i = 1, \ldots 5)$, as described in the previous section. A combination of these input and output data forms a sample point in the training and the validation data, and we need to generate many independent combinations to prepare the training and the validation data.

The procedure for sampling the M-R points is sketched in figure 4. We follow the basic strategy described in section 2.3 and introduce minor modifications for the current setup. For each randomly generated EoS, $p = p(\varepsilon)$, we solve the TOV equation to get the M-R curve and identify the maximum mass $M_{\max}$ of the neutron star, corresponding to (1) and (2) in figure 4. If $M_{\max}$ does not reach the observed maximum, such an EoS is rejected from the ensemble. Then, for each EoS, we randomly sample 15 pairs of (M_i, R_i) on the corresponding M-R curve, as shown in (3) in figure 4, assuming a uniform distribution of M_i over the interval $[M_{\odot}, M_{\max}]$. If there are multiple values of R corresponding to one M, we take the largest R discarding others belonging to unstable branches. In this way we select 15 points of (M_i^*, R_i^*) on the M-R curve.

We also expect the NN to learn that observational data should include uncertainties, ΔM_i and ΔR_i. We randomly generate ΔM_i and ΔR_i according to the normal distribution with assumed variances, $\sigma_M = 0.1 M_{\odot}$ for the mass and $\sigma_R = 0.5\,\mathrm{km}$ for the radius. These variances should be adjusted according to the real observational uncertainty as we will treat in section 4, but for the test purpose in this section, we simplify the treatment by fixing σ_M and σ_R by hand. Furthermore, the distributions are not necessarily centered on the genuine M-R curve as in (3) in figure 4. So, we shift data points from (M_i^*, R_i^*) to $(M_i = M_i^* + \Delta M_i, R_i = R_i^* + \Delta R_i)$. Importantly, we repeat this procedure to generate

Layer index	Neurons	Activation Function
0	30	N/A
1	60	ReLU
2	40	ReLU
3	40	ReLU
4	5	tanh

Table 1. Neural network architecture used in section 3. In the input layer, 30 neurons correspond to input 15 points of the mass and the radius. In the last layer 5 neurons correspond to 5 output parameters of the EoS.

an ensemble of randomly shifted data for each EoS, and the repeating number (and thus the number of obtained shifted data) is denoted by n_s in this work. Our typical choice is $n_s = 100$, whose effects on learning will be carefully examined in later discussions. For studies in this section, we have prepared 1948 EoSs (2000 generated and 52 rejected), so the total size of the training data set is $1948 \times n_s$. We find that the learning proceeds faster if the data is normalized appropriately to be consistent with the Glorot initialization; we use $M_i/M_{\rm norm}$ and $R_i/R_{\rm norm}$ with $M_{\rm norm} = 3M_\odot$ and $R_{\rm norm} = 20\,\text{km}$.

The NN is optimized to fit the training data so as to have a predictive power. To test it, we use the validation loss calculated for the validation data, i.e., independent mock data of the neutron star observation. We separately generate 200 EoSs, among which 196 EoSs pass the two-solar-mass condition, and sample an $n_s = 1$ observation for each EoS to generate the validation data set.

It is worth mentioning here that calculating M-R curves requires repeated numerical integration of the TOV equation with various boundary conditions, which is computationally expensive. Generally speaking, in the machine learning approach, the preparation of the training data is often the most time-consuming part. In contrast, increasing the sample size by repeating the observation procedure does not need further computational cost. This procedure can actually be regarded as a kind of the data augmentation by the noise injection, where the observational errors play the role of the injected noise. Let us call this procedure *observational data augmentation* hereafter. This observational data augmentation has actually another virtue, which will be discussed in later sections.

We are constructing a NN that can give us one EoS in the output side in response to one "observation" of 15 neutron stars, $(M_i,\ R_i)$ $(i = 1, \ldots, 15)$ in the input side. Thus, the number of neurons in the input and the output layers should be matched with the size of the observational data and the number of EoS parameters, respectively. The input layer with 30 neurons matches 15 M-R points (30 input data). We sorted 15 M-R pairs by their masses in ascending order, but we found that the sorting does not affect the overall performance. The output neurons in the last layer correspond to the prediction target, i.e., 5 parameters of the speed of sound.

The concrete design of our NN is summarized in table 1. We choose the activation function at the output last layer as $\sigma^{(4)}(x) = \tanh(x)$ so that the speed of sound is automatically bounded in a causal range $c_s^2 < 1$. In principle the output could be unphysical:

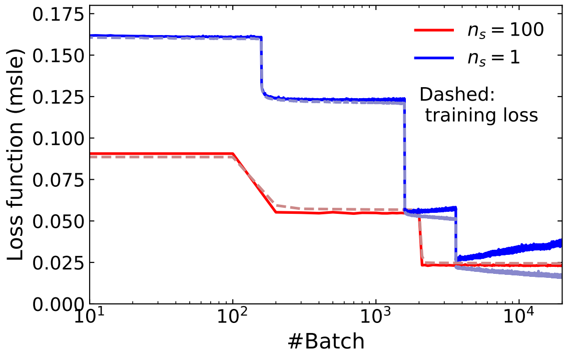


Figure 5. Typical behavior of learning curves in our setup. Solid lines show the validation loss and the dashed lines show the training loss.

$c_s^2 < 0$ may occur, and then we readjust it as $c_s^2 = \delta$. For the other layers we choose the ReLU, i.e., $\sigma^{(\ell)}(x) = \max\{0, x\}$ ($\ell = 1, \ldots, L-1$), which is known to evade the vanishing gradient problem.

3.2 Learning curves with the observational data augmentation

The stochastic gradient descent method is an iterative procedure to adjust the NN parameters step by step and to gradually make the loss function smaller. To judge the point where we stop the iteration, we need to monitor the convergence of the loss function in training. For this purpose we plot the loss function as a function of *training time* (i.e., the number of iteration units), which we call the *learning curve* in this work.

The actual shape of the learning curve depends on the initial parameters of the NN and also on the choice of mini-batches in the stochastic gradient descent method. Although the learning curve differs every time we start the new training, we can still find some common tendency. In figure 5 we show the typical behavior of learning curves for our neutron star problem. It should be noted that the horizontal axis in figure 5 is the number of mini-batches. In this study we will compare the training speeds with different n_s but with the fixed mini-batch size. The number of epochs — the number of scans over the whole training data set — is often used as the unit of training time, but in the present study for the training speeds, we need to adopt the number of mini-batches which is proportional to the actual number of parameter updating and the computational time of the training.

In figure 5 we plot two different learning curves for a single training: one for the *training loss* (dashed line) and the other for the *validation loss* (solid line). The training and the validation losses are the loss functions calculated for the training and the validation data sets, respectively, by using eq. (2.13). The former diagnoses how well the model *remembers* the training data, and the latter diagnoses how well the model properly *predicts* unseen data by generalizing the learned information.

Here, we compare the learning curves for two different training data sets with $n_s = 100$ and $n_s = 1$. By observing the plot like figure 5 for many trials on the training, we can read out two general tendencies:

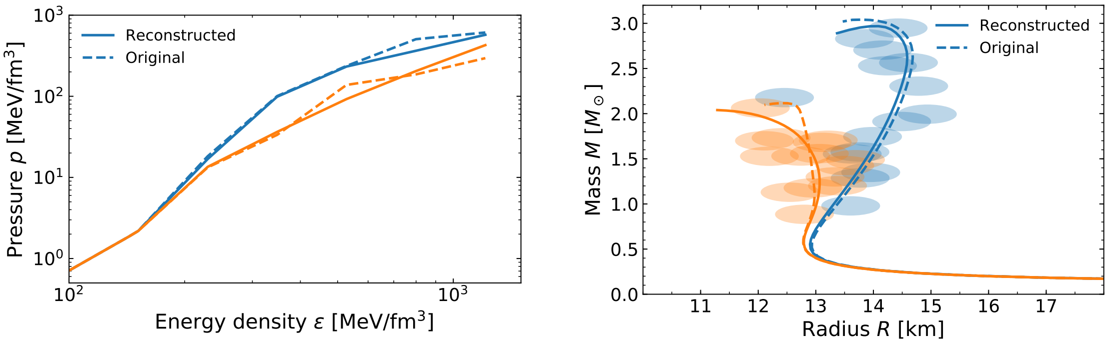


Figure 6. (Left) Two typical examples of the randomly generated EoSs (dashed line) and the NN outputs (solid lines) reconstructed from one observation of 15 M-R points [see the right panel for actual $(M_i,\ R_i)$]. (Right) Randomly sampled 15 data points (centers of the ellipses) and the M-R curve from the reconstructed EoS (solid lines) and the M-R curve from the original EoS (dashed lines). The orange and blue colors correspond to two EoSs shown in the same color in the left panel.

1. The value of validation loss of $n_s = 1$ is larger than that of $n_s = 100$. A possible interpretation is that the loss function can be more trapped in local minima/saddle points for smaller n_s.

2. For the number of batches $\gtrsim 10^3$, for $n_s = 1$, the training loss keeps decreasing with increasing number of batches, whilst the validation loss turns to increasing behavior. The loss function of $n_s = 1$ apparently shows overfitting behavior but the overfitting is not seen for $n_s = 100$.

From these results, we may argue that the data augmentation with sufficiently large n_s could be useful to improve the problems of local minimum trapping and overfitting. To justify or falsify this speculation, we will perform further quantitative analyses on the observational data augmentation under a simple setup in section 5. We note in passing that recent developments show a possibility of over-parametrized networks, which has a very large number of parameters, to reconcile a good generalization ability and an elusion of overfitting [94, 95]. Here, our point is that the data augmentation is needed to incorporate the physical uncertainty rather than to improve the learning quality, but it may have such a byproduct.

3.3 Typical examples — EoS reconstruction from the neural network

Once the loss function converges, we can use the trained NN model to infer an EoS from an observation of 15 M-R points. We picked two typical examples represented by orange and blue in figure 6. In the left panel of figure 6 the dashed lines represent randomly generated EoSs. We note that two EoSs are identical in the low density region because the SLy4 is employed at $\varepsilon \leq \varepsilon_0$. The corresponding genuine M-R relations are shown by the dashed lines in the right panel of figure 6. Randomly sampled mock observations consisting of 15 M-R points are depicted by ellipses where the centers are the "observed"

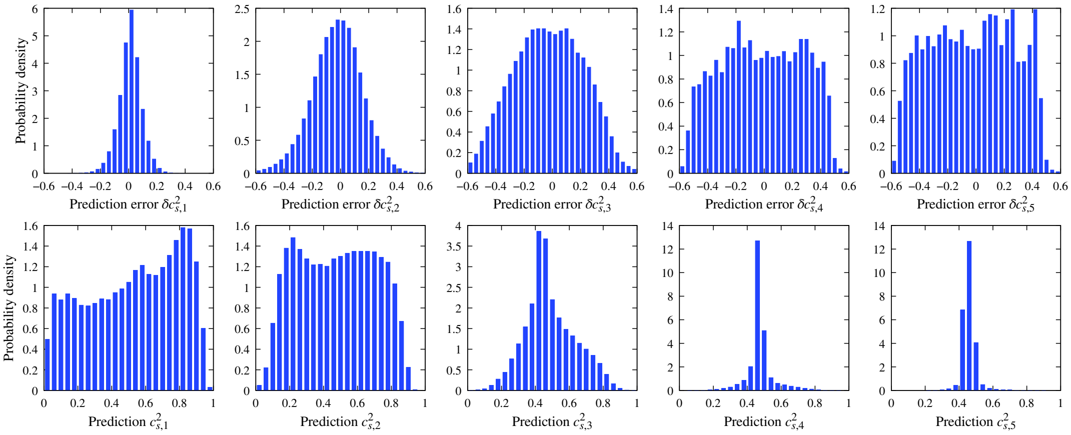


Figure 7. The upper and the lower panels show the histograms of prediction errors and values, respectively, of $c^2_{s,i}$. The vertical axis is the probability density, i.e., the count in each bin is divided by the bin width and the total count.

(M, R) and the major and minor axes show observational uncertainties in the R and M directions, respectively. The reconstructed EoSs are depicted by solid lines in the left panel of figure 6. We can see that the reconstructed EoSs agree quite well with the original EoSs for these examples. It would also be interesting to make a comparison of the M-R relations corresponding to the original and the reconstructed EoSs. The solid lines in the right panel of figure 6 represent the M-R relations calculated with the reconstructed EoSs. As is expected from agreement in the EoSs in the left panel of figure 6, the original and the reconstructed M-R relations are close to each other. More details are already reported in our previous publication [81].

3.4 Error correlations

We make the detailed error analysis of the reconstructed EoSs in terms of the speed of sound. To increase the statistics, for the present analysis, we set $n_s = 1000$ (remember that increasing n_s is almost costless).

Now, we introduce the prediction error in the speed of sound:

$$\delta c^2_{s,i} \equiv (c^2_{s,i})^{(\text{rec})} - (c^2_{s,i})^* \quad (i = 1, \dots, 5)\,, \tag{3.1}$$

where $(c^2_{s,i})^{(\text{rec})}$ and $(c^2_{s,i})^*$ are the reconstructed and original values, respectively, for the speed of sound. The first row of figure 7 shows the histograms of $\delta c^2_{s,i}$ $(i = 1, \dots 5)$. We see that the prediction error at the lower density (i.e., smaller i) is centered at the origin while that at the higher density (i.e., larger i) is distributed widely, which means that the observational data efficiently constrains lower density regions. At the highest density with $i = 5$, the prediction error roughly follows the uniform distribution within the physical range. This behavior can be understood as follows. The original values of the speed of sound in the validation data are generated by the uniform distribution in the range $[0, 1]$ while the distribution of the predicted $c^2_{s,i}$ $(i = 1, \dots 5)$ are shown in the second row of figure 7. At lower density, the distribution of the prediction roughly follows the original uniform distribution. However, at higher density, the distribution of the prediction peaks around $c_{s,i}^2 \sim 0.5$. At the highest density, therefore, the NN always predicts $c_{s,i}^2 \sim 0.5$ for any inputs. This is because $c_{s,i}^2 \sim 0.5$ is an optimal prediction to minimize the loss function when the constraining information is inadequate. The exact peak position is slightly smaller than 0.5, which is because our choice of the loss function, `msle`, gives larger weights for smaller values. Thus, the approximate uniform distribution of the prediction errors at the highest density (as seen in the upper rightmost figure in figure 7) just reflects the distribution of the original values relative to the fixed prediction.

From this analysis we conclude that the observation data should contain more information on the lower density EoS but not much on the high density EoS around $\varepsilon \sim 8\varepsilon_0$. This is quite natural, but such a quantitative demonstration in figure 7 is instructive. Also, the results in figure 7 tell us two important aspects of our NN approach. First, the NN tends to give a moderate prediction, not too far from all possible values, when the uncertainty is large. This is nontrivial: the posterior distribution of EoS is random, but such information is lost after the NN filtering. This is reasonable behavior in a sense that we want to design the NN to predict the most likely EoS instead of the posterior distribution. Secondly, the moderate prediction for unconstrained cases may explicitly depend on the choice of the loss function and also on the prior distribution, i.e., the distribution of the EoSs in the training data set in our problem. Thus, we should be careful of possible biases when we discuss unconstrained results in the high density regions.

The error analysis so far is separately made for each EoS parameter. In figure 8, to extract more detailed information, we plot the marginalized distributions for different pairs of the EoS parameters with corresponding Pearson's correlation coefficients r_{ij}. We see that the speeds of sound of neighboring density segments have negative correlations, which is similar to the behavior seen in the Bayesian analysis [63]. The underlying mechanism for this behavior can be explained in the same way as in ref. [63]: an overprediction in a segment can be compensated by an underprediction in the adjacent segment, and vice versa. The correlations are clearer in the lower density regions in which predicted $c_{s,i}^2$'s have better accuracy. We also observe a small and positive correlation, e.g., $r_{13} > 0$, between the parameters separated by two segments. This is also expected from two negative correlations in between. The other correlations involving parameters in the high density regions are consistent with zero within the statistical error.

3.5 Conventional polynomial regression

It is an interesting question how the machine learning method could be superior to other conventional approaches as a regression tool. In this section we make a detour, introducing a polynomial model, and checking performance of a traditional fitting method. We can model the regression of EoS from the M-R data with the following polynomial fitting of degree n:

$$\begin{aligned} y_i &= g_i(\{x_j\}|\{a_{ij_1j_2\cdots j_N}\}) \\ &\equiv \sum_{\substack{j_k\geq 0\ (k=1,\ldots,N) \\ j_1+j_2+\cdots+j_N\leq n}} a_{ij_1j_2\cdots j_N} x_1^{j_1} x_2^{j_2} \cdots x_N^{j_N}\,, \end{aligned} \tag{3.2}$$

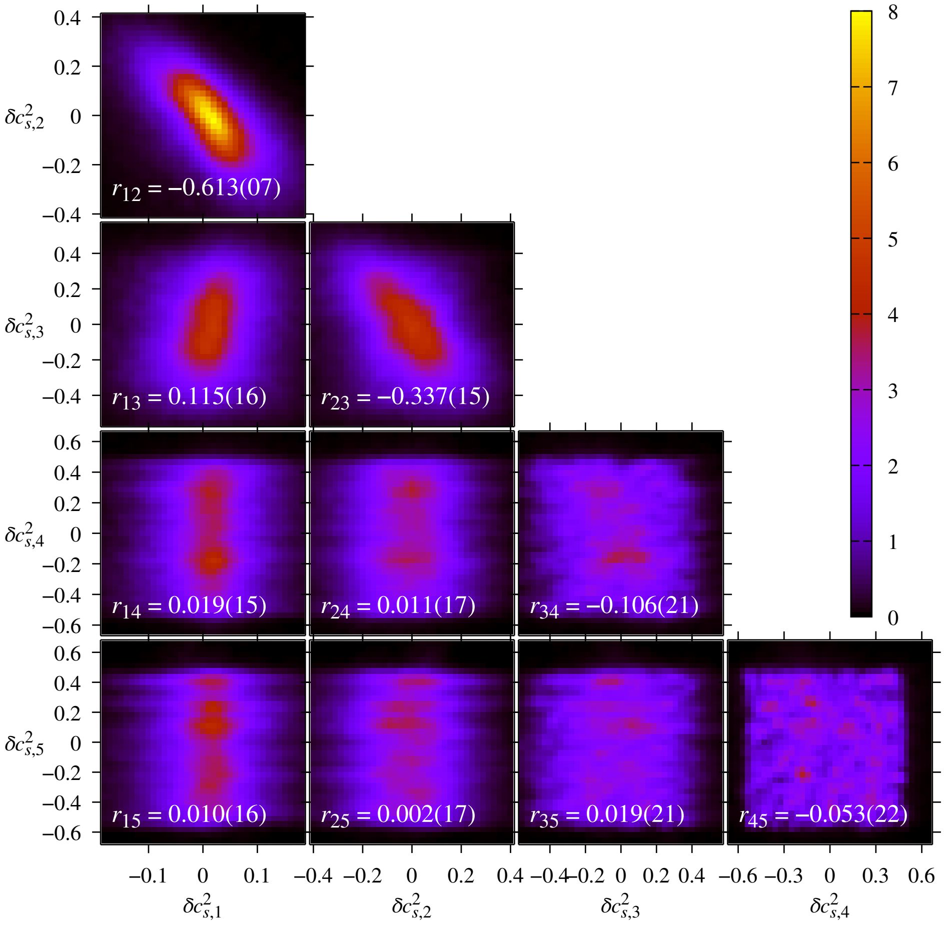


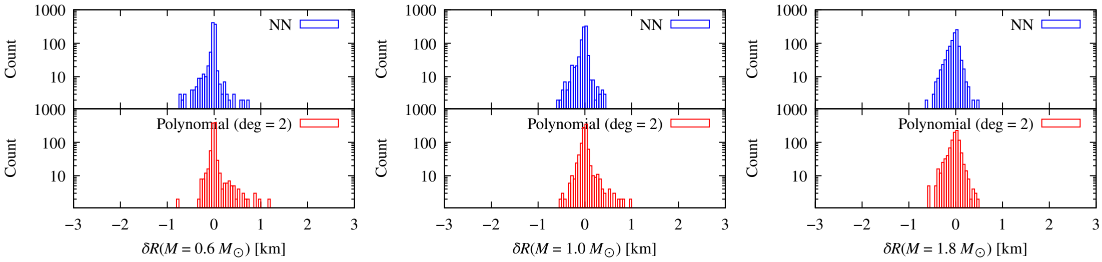
Figure 8. Predicted errors marginalized for different pairs of EoS parameters. The first to fourth columns correspond to $\delta c^2_{s,i}$ ($i = 1, \dots 4$). Similarly, the first to fourth rows correspond to $\delta c^2_{s,i}$ ($i = 2, \dots 5$). Colors show the normalized counts in each bin, i.e., the counts in the bin divided by the total count and multiplied by the bin number. Pearson's correlation coefficients r_{ij} for respective parameter pairs are shown in panels with the statistical uncertainty enclosed in parentheses.

where $\{x_i\}$ and $\{y_i\}$ are the input and the output of the training data set, respectively, with $N = 30$ being the size of the input vector $\{x_i\}$, and $\{a_{ij_1j_2\cdots j_{N_1}}\}$ are the fitting parameters. For the optimization we employ the traditional least-square method with the loss function in eq. (2.12) specified as `mse`, i.e., the mean square errors:

$$\ell_{\texttt{mse}}(\boldsymbol{y}, \boldsymbol{y}') \equiv |\boldsymbol{y} - \boldsymbol{y}'|^2 \,. \tag{3.3}$$

In this case the mapping is linear with respect to the fitting parameters, which means the problem is essentially the (multivariate) linear regression. Thus, we can find an analytical solution of the exact global minimum of the loss function (3.3) in this traditional method.

Here we comment on the number of parameters in the polynomial model, which is given by the multiset coefficient: $n_{\text{param}} = \left(\!\binom{N+1}{n}\!\right) = \binom{N+n}{n}$. For $N = 30$ in the present problem, as long as $n \ll N$, we can approximate $n_{\text{param}} = \mathcal{O}(N^n)$ which explodes as the



(a) (b) (c)

Figure 9. Histograms of δR from the NN (upper) and the polynomial (lower) regression for (a) $0.6M_\odot$, (b) $1.0M_\odot$, and (c) $1.8M_\odot$.

degree n gets larger. Explicit calculations read: $n_{\text{param}} = 31$ ($n = 1$), 496 ($n = 2$), 5456 ($n = 3$), 46376 ($n = 4$), etc. The drawback of the linear regression is a huge computational cost of $\mathcal{O}(n_{\text{param}}^2 |\mathcal{D}|)$.

3.6 Comparison between the neural network and the polynomial regression

In section 3.3 we already observed two successful examples of the EoS reconstruction. For other EoSs in the validation data, the reconstructed M-R curves agree well with the genuine ones. In this section we shall quantify the overall agreement, and for this purpose we enlarge the size of the validation data; namely, we randomly generate 1000 EoSs, among which 967 EoSs pass the two-solar-mass condition, and we carry out statistical analyses with 967 EoSs.

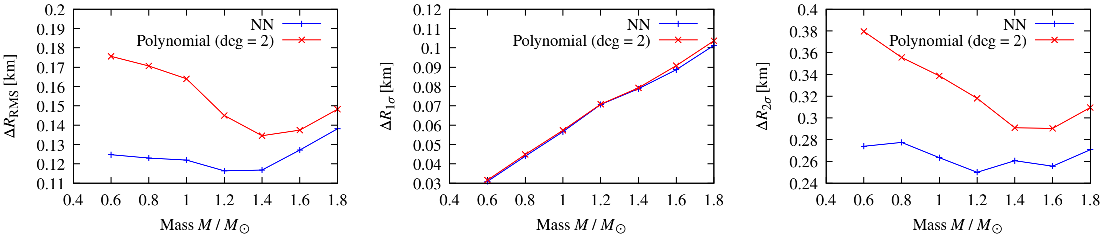
We define the overall reconstruction accuracy by calculating the radius deviation, i.e., the distance between the solid and dashed lines as already shown in the right panel of figure 6:

$$\delta R(M) \equiv R^{(\text{rec})}(M) - R^*(M)\,, \tag{3.4}$$

where $R^{(\text{rec})}$ and $R^*(M)$ are the reconstructed radius (solid) and the genuine original radius (dashed), respectively, at the mass M. Then, we estimate the root mean square (RMS) of radius deviations (3.4) using 967 data at masses ranging from 0.6 $M_\odot$ to 1.8 $M_\odot$ with an interval of 0.2 $M_\odot$. We calculate $\delta R(M)$ for both the NN and the polynomial regression to make a quantitative comparison.

The logarithmic histogram of $\delta R(M)$ is plotted in figure 9 for $M = 0.6M_\odot$ [as shown in (a)] $1.0M_\odot$ [as shown in (b)], and $1.8M_\odot$ [as shown in (c)]. In the plot of the logarithmic probability density the Gaussian distribution or the normal distribution takes a quadratic shape, so the δR histograms from both the NN and the polynomial regression have a strong peak at $\delta R = 0$ for all M. The tails are wider than the normal distribution, and the polynomial results (lower panels) exhibit even slightly wider tails than the NN results (upper panels).

We can look into this tendency in more quantitative manner in figure 10. In the leftmost (a) of figure 10 we plot the RMS of $\delta R(M)$, i.e., ΔR_{RMS}. We see that ΔR_{RMS} from the NN is around $\sim 0.12\,\text{km}$ for a wide range of masses. This indicates that our NN



(a) (b) (c)

Figure 10. Overall performance measured with (a) ΔR_{RMS}, (b) $\Delta R_{1\sigma}$, and (c) $\Delta R_{2\sigma}$.

method works surprisingly good; remember that data points have random fluctuations by $\Delta R \sim 0.5\,\text{km}$. It should be noticed that, even without neutron stars around $M = 0.6$–$0.8 M_\odot$ in mock M-R data of our setup, the RMS of the corresponding radii are still reconstructed within the accuracy $\sim 0.1\,\text{km}$. One might think that this should be so simply because we use the SLy4 in the low energy region, but in view of figure 6, different EoSs lead to significantly different radii even in this region of $M = 0.6$–$0.8 M_\odot$. Figure 10 (a) clearly shows that the NN results surpass the polynomial outputs with $n = 2$.

When the distribution has long tails, as seen in figure 9, the RMS may not fully characterize the data behavior. To quantify the deviation of the reconstruction more differentially, we can define quantile-based widths, $\Delta R_{1\sigma}$ and $\Delta R_{2\sigma}$, corresponding to $\simeq 68\%$ and $\simeq 95\%$ of δR contained within the half widths, respectively. They can be explicitly expressed as

$$\Delta R_{\alpha\sigma} \equiv \frac{1}{2}\left[F_{\delta R}^{-1}\left(\frac{1+\operatorname{erf}\frac{\alpha}{\sqrt{2}}}{2}\right) - F_{\delta R}^{-1}\left(\frac{1-\operatorname{erf}\frac{\alpha}{\sqrt{2}}}{2}\right)\right], \tag{3.5}$$

where $F_{\delta R}^{-1}(p)$ is the quantile function of the δR distribution, i.e., the value of δR at the fraction p integrated from small δR. The error function, $\operatorname{erf}(\alpha/\sqrt{2}) \equiv (1/\sqrt{2\pi}) \int_{-\alpha}^{\alpha} dx e^{-x^2/2}$, is used to translate the statistical significance by σ to the probability, e.g., $\operatorname{erf}(1/\sqrt{2}) \simeq 68\%$ and $\operatorname{erf}(2/\sqrt{2}) \simeq 95\%$. If δR obeys the normal distribution $\mathcal{N}(0, \sigma^2)$, in general, $\Delta R_{\alpha\sigma} = \alpha\sigma$ follows. Therefore, we can use $\Delta R_{\alpha\sigma}/\alpha$ as alternatives to $\Delta R_{\text{RMS}} = \sigma$. For more general distributions it is important to note that these quantile-based widths have a clear meaning as the 68% and 95% confidence intervals unlike ΔR_{RMS}.

In figures 10 (b) and (c) we plot $\Delta R_{1\sigma}$ and $\Delta R_{2\sigma}$. It is interesting that the $\Delta R_{2\sigma}$ results appear very different while the $\Delta R_{1\sigma}$ results are almost indistinguishable between the NN and the polynomial results. The polynomial results are consistently larger than the NN results. This behavior of $\Delta R_{2\sigma}$ implies that the polynomial results may be contaminated by outliers of wrong values far from the true answer. We can understand this by relating it to Runge's phenomenon of the polynomial approximation; higher order terms can cause large oscillation in the interpolation and the extrapolation. In particular, the extrapolation may easily break down by higher order terms that blow up for large inputs. In contrast, the NNs with ReLU activation only contain up to the linear terms so that the outputs are stable.

4 EoS estimation from the real observational data

In the previous section we have studied our methodology to infer the most likely EoS from observational data and quantitatively confirmed satisfactory performance to reproduce the correct results. Now, we shall apply the method to analyze the real observational data and constrain the real neutron star EoS.

For practical purposes it is important to develop a way to estimate uncertainties around the most likely EoS within the framework of the machine learning method. As we have already discussed, the EoS inference from the M-R data is an ill-posed problem and the solution cannot be uniquely determined. Thus, the prediction from the NN method cannot avoid uncertain deviations from the true EoS. To employ a predicted EoS as a physical output, we need to quantify the uncertainty bands on top of the results. In this section we consider two different approaches for the uncertainty quantification.

4.1 Compilation of the neutron star data

For the real data we use the M-R relation of the neutron stars determined by various X-ray observations. There are three sorts of sources we adopt here: 8 neutron stars in quiescent low-mass X-ray binaries (qLMXB) [2, 37, 38], 6 thermonuclear bursters with better constraints than the qLMXBs [2, 37] as well as a rotation-powered millisecond pulsar, PSR J0030+0451, with thermal emission from hot spots on the stellar surface [43, 44]. The data from the first two sources is tabulated in refs. [2, 37, 38], especially in figure 4 of ref. [2].[2] The data from the last source is recently investigated in the NICER mission, and there are two independent analyses based on the same observation, among which here we use the data given in ref. [43]. All of these M-R relations are provided in the form of the Bayesian posterior distribution, which takes the two-dimensional probability distribution on the R-M plane (see also figure 1 in ref. [82] for a graphical representation of the probability distribution).

Ideally, with sufficient computational resources, the machine learning would be capable of directly dealing with such two-dimensional probability distribution using, for example, the convolutional neural network (CNN). In the present work, however, we simplify our analysis by approximately characterizing a two-dimensional probability distribution with four parameters following the prescription proposed in our previous publication [82]: the means and the variances of the marginal distributions with respect to M and R. More specifically, we make projection of the two-dimensional distribution onto the one-dimensional M-axis (and R-axis) by integrating over R (and M, respectively). In this way we marginalize the two-dimensional distribution with respect to M and R into two one-dimensional distributions, which are eventually fitted by the Gaussians with the mean and the variance.

4.2 Training data generation with observational uncertainty

We should revise the data generation procedure previously illustrated in figure 4 in section 3.1. Previously, for the mock data analysis, we assumed the universal variances,

[2] The data is distributed at the website: http://xtreme.as.arizona.edu/NeutronStars/.

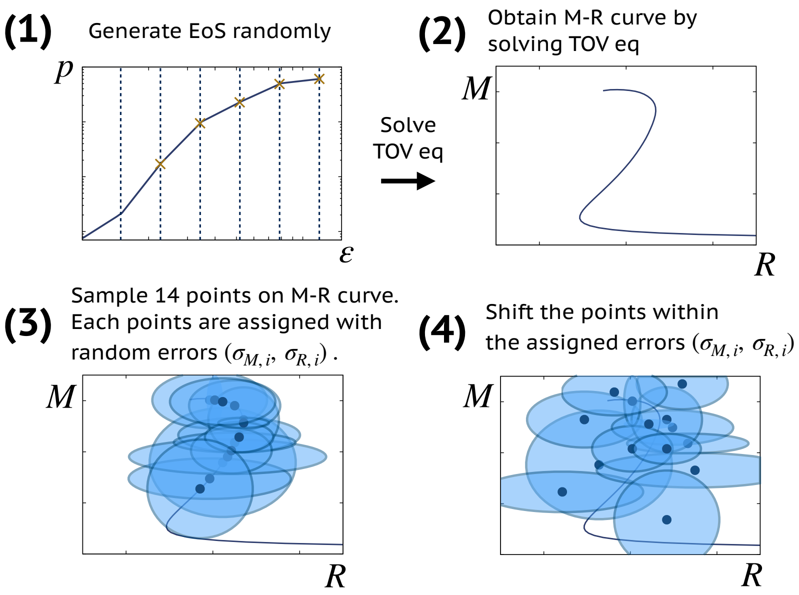

(1)
Generate EoS randomly
p
ε
Solve
TOV eq
(2)
Obtain M-R curve by
solving TOV eq
M
R
(3)
Sample 14 points on M-R curve.
Each points are assigned with
random errors $(\sigma_{M,i}, \sigma_{R,i})$.
M
R
(4)
Shift the points within
the assigned errors $(\sigma_{M,i}, \sigma_{R,i})$
M
R

Figure 11. Schematic flow of data generation procedure for the analysis in section 4.

$\sigma_M = 0.1M_\odot$ and $\sigma_R = 0.5\,\text{km}$, for all ΔM_i and ΔR_i. In reality, however, they should vary for different i, i.e., $\sigma_{M,i}$ and $\sigma_{R,i}$ should correspond to the observational uncertainties of (M_i, R_i). To deal with the real observational data, the revised procedure for sampling the M-R points is sketched in figure 11. We need to design the NN with an input of information including $\sigma_{M,i}$ and $\sigma_{R,i}$: the input variables are extended to $(M_i,\ R_i;\ \sigma_{M,i},\ \sigma_{R,i})$.

We shall recapitulate the data generation scheme as follows. In the same way as in section 3 we prepare 5 EoS parameters $c^2_{s,i}$ $(i = 1, \dots 5)$ in the output side. In this section the training data comprises 14×4 input parameters, i.e., $(M_i,\ R_i;\ \sigma_{M,i},\ \sigma_{R,i})$ $(i = 1, \dots, 14)$. We note that i runs to not 15 but 14 corresponding to the number of observed neutron stars as explained in section 4.1. We calculate the M-R curve for each EoS, and then select 14 points of (M^*_i, R^*_i) on the M-R curve and add statistical fluctuations of ΔM_i and ΔR_i [see figure 11 (3)]. Let us go into more detailed procedures now. Unlike σ_M and σ_R in section 3 here we randomly generate $\sigma_{M,i}$ and $\sigma_{R,i}$ differently for $i = 1, \dots 14$. These variances, $\sigma_{M,i}$ and $\sigma_{R,i}$, are sampled from the uniform distributions, $[0, M_\odot)$ and $[0, 5\ \text{km})$, respectively. In view of the observational data, these ranges of the distributions should be sufficient to cover the realistic situations. Then, ΔM_i and ΔR_i are sampled according to the Gaussian distributions with these variances, $\sigma_{M,i}$ and $\sigma_{R,i}$. Finally we obtain the training data, $(M_i = M^*_i + \Delta M_i, R_i = R^*_i + \Delta R_i; \sigma_{M,i}, \sigma_{R,i})$ $(i = 1, \dots 14)$ [see figure 11 (4)]. Hereafter we call these 14 tetrads of $(M_i, R_i; \sigma_{M,i}, \sigma_{R,i})$ an *observation*.

Now we prepare the training data set by taking multiple observations. For each EoS we randomly generate 100 different pairs of $(\sigma_{M,i}, \sigma_{R,i})$, and then we make another 100 observations for each $(\sigma_{M,i}, \sigma_{R,i})$. From the former 100 pairs the NN is expected to learn that the observational uncertainties may vary, and the latter tells the NN that the genuine M-R relation may deviate from the observational data. In total we make $n_s = 10000$

Layer index	Neurons	Activation Function
0	56	N/A
1	60	ReLU
2	40	ReLU
3	40	ReLU
4	5	tanh

Table 2. Neural network architecture used in section 4. In the input layer, 56 neurons correspond to the input 14 points of the mass, the radius, and their variances. The design of the other layers is kept the same as in table 1.

($= 100 \times 100$) "observations" per one EoS. The size of the whole training data set is thus 100 times larger than before.

We modify the architecture of the NN used in this section accordingly. The number of neurons in the input layer becomes $56\,(= 4 \times 14)$ (the performance test was done with 15 points, but we have numerically confirmed that the same level of performance is achieved with 14 points as well). Since we already know from the mock data analysis that the mass sorting does not affect the performance, we keep the mass ordering as generated randomly, unlike in section 3.1. We normalize the input data as $M_i/M_{\rm norm}$, $R_i/R_{\rm norm}$, $\sigma_{M,i}/\sigma_{M\,{\rm norm}}$, and $\sigma_{R,i}/\sigma_{R\,{\rm norm}}$ with $M_{\rm norm} = 3M_\odot$, $R_{\rm norm} = 20\,{\rm km}$, $\sigma_{M\,{\rm norm}} = 1M_\odot$, and $\sigma_{R\,{\rm norm}} = 5\,{\rm km}$. Aside from the input layer, the NN design of the other layers is chosen to be the same as before, as summarized in table 2. For the NN optimization we adopt Adam, the same as in section 3, but with different mini-batch size, 1000. Incorporating the observational uncertainties the training data set becomes larger as compared to before, and we expect that a larger mini-batch size would fasten the training.

4.3 Two ways for uncertainty quantification

Here we prescribe two independent methods to quantify uncertainties in the output EoS based on different principles.

The first one utilizes the validation data, and the procedure is similar to that in section 3.6. The basic idea is that, once we have trained the NN, we can evaluate the prediction accuracy from the validation data. We generate 100 samples of the validation data set whose input variances are chosen to be in accord with the real observational uncertainties as explained in section 4.1. For the validation data, we know the true M-R relation so that we can evaluate the deviation $\delta R(M)$ as defined in eq. (3.4). Using the whole validation data set, we calculate the root-mean-square deviation, $\Delta R_{\rm RMS}$, and regard it as the systematic error inherent in the NN approach. Later we show this uncertainty by the label "validation" as seen in figure 13.

The second one is the ensemble method in machine learning. This method is usually used to enhance the stability and performance of the predicted output from NNs. Here we repurpose it to quantify uncertainty of the predicted output. The idea is concisely summarized in figure 12. In this method we set up multiple NNs independently: NN_1, $\mathrm{NN}_2, \ldots,\mathrm{NN}_N$ with N being the number of prepared NNs. We perform random

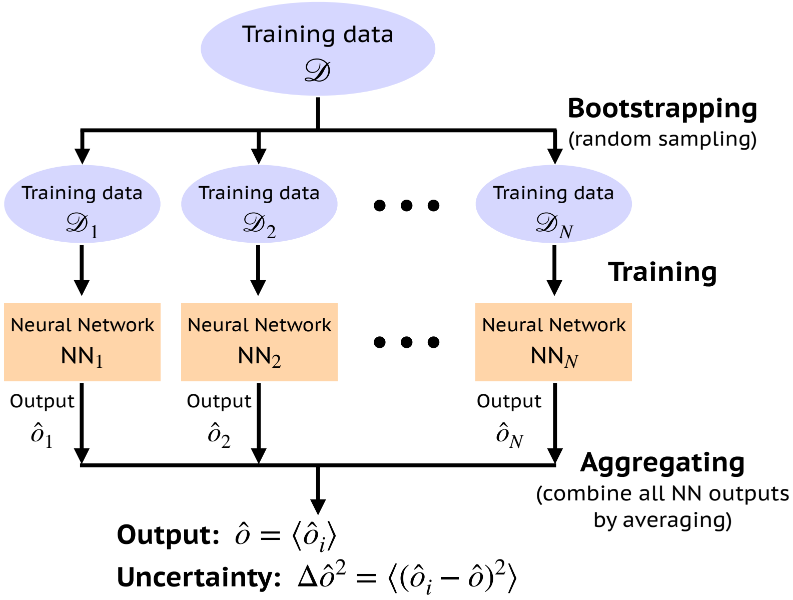


Figure 12. Illustration of the ensemble method procedure for uncertainty quantification.

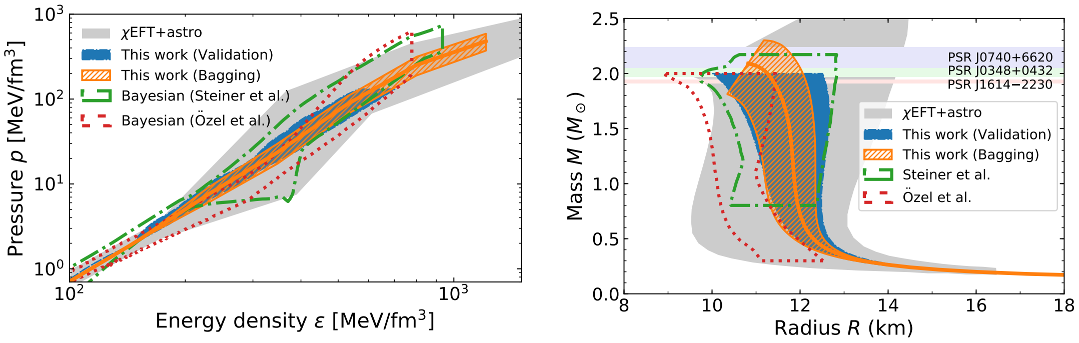
sampling from $\mathcal{D}$ to generate different subsets, $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_N$ and train each NN using $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_N$. This random sampling is commonly referred to as *bootstrapping*. After the training, by feeding the input data, each NN predicts output values, which we symbolically denote by $\hat{o}_1, \hat{o}_2, \ldots, \hat{o}_N$ as in figure 12. Finally, aggregating the outputs from these multiple NNs, i.e., averaging all the outputs, we get the overall output $\hat{o} = \langle \hat{o}_i \rangle \equiv N^{-1} \sum_i \hat{o}_i$. Here we can also calculate the variance by $\Delta\hat{o}^2 = \langle (\hat{o}_i - \hat{o})^2 \rangle$, and regard it as "uncertainty". This whole procedure, comprising bootstrapping and aggregating, is named *bagging*[3] [96]. In this work, we choose $N = 10$. If some regions of the EoS are insensitive to the M-R observation, independently trained 10 NN models would lead to different EoSs in such unconstrained regions. From $\Delta\hat{o}$ that quantifies how much 10 output EoSs vary, we can estimate the uncertainty around the output $\hat{o}$. We use this bagging for most of the analyses as shown by the band labeled by "Bagging" in figures 13.

We have introduced the two natural ways to quantify uncertainties, but our working prescriptions are still to be developed. In fact there is no established method yet. A more systematic way for the uncertainty quantification might be possible. This is an interesting problem left for future research in the general context of machine learning.

4.4 The most likely neutron star EoS

In figure 13 the orange line is the most likely neutron star EoS deduced from our NN approach. We estimated uncertainty from the bagging (shown by the band with a hatch pattern labeled by "bagging") and the validation (shown by the blue band labeled by "validation"). We plot bands from other works in the figure for comparison. The gray band

[3] In the strict sense, the term "bagging" is used for the random sampling procedure *with* replacement of data. Here we use "bagging" even without replacement in a loose sense.

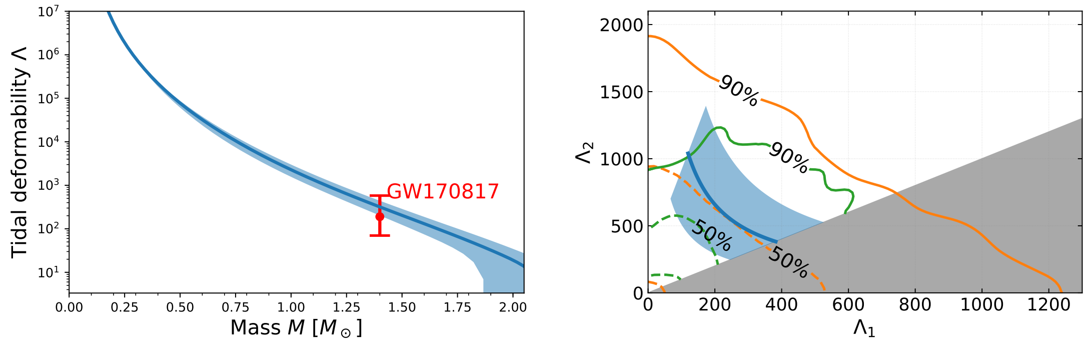


(a) (b)

Figure 13. (a) EoSs deduced from the observational M-R data of qLMXBs and thermonuclear bursters. The shaded blue and hatched orange bands represent our 68% credibility bands from the validation and the bagging estimations. The χEFT prediction and the Bayesian results (Steiner et al. [35, 36] and Özel et al. [2, 37, 38]) are overlaid for reference. The former band represents 68% CL, while the latter shows the contour of e^{-1} of the maximum likelihood. (b) M-R relations corresponding to the deduced EoSs from this work with references to other approaches.

represents an estimate from the χEFT calculation combined with polytropic extrapolation and the two-solar-mass pulsar constraint [97] (labeled by "χEFT+astro"). Because χEFT is an *ab initio* approach, any reasonable predictions should lie within the gray band, and indeed our results are found to be near the middle of the gray band. The preceding Bayesian analyses [2, 35–38] are also overlaid on figure 13. While Özel et al. [2, 37, 38] and our present analysis use the same astrophysical data, Steiner et al. [35, 36] employs a subset of the data, i.e., 8 of X-ray sources. One may think that our prediction gives a tighter constraint than the others, but the narrowness of the band may be related with the implicit assumption in our EoS parametrization; we will come back to this point in section 4.5 (see figure 17). Figure 13 (b) shows the M-R curves corresponding to the EoSs in (a). We see that our EoS (blue curve) certainly supports neutron stars with $M > 2M_\odot$ [30–33].

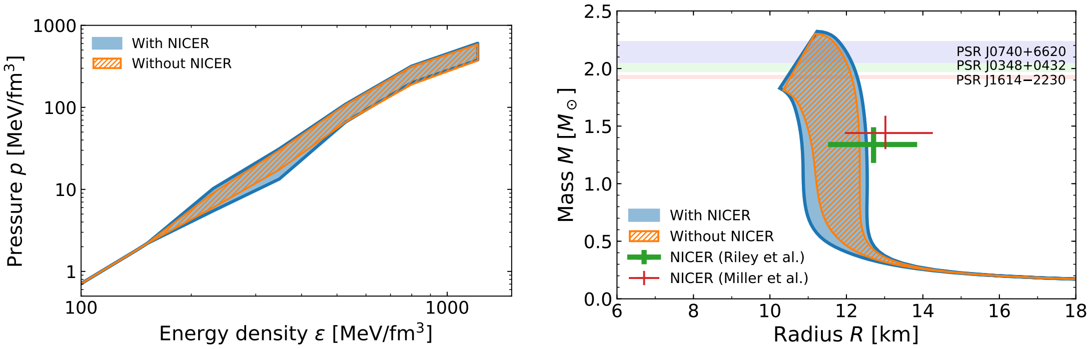
Figures 14 (a) and (b) show the tidal deformability and their correlation, respectively, in the binary neutron star merger GW170817. Once an EoS is given, the dimensionless tidal deformability, Λ, results from a quantity called the Love number k_2, which is derived from the Einstein equation under static linearized perturbations to the Schwarzschild metric due to external tidal fields. Practically, we solve a second-order ordinary differential equation in combination with the TOV equation; see refs. [98, 99] for the explicit form of the equations. The blue band in figure 14 (a) represents Λ from the EoS we inferred in the present work, which is consistent with the merger event GW170817 indicated by the red bar. In figure 14 (b) we show the correlation of the tidal deformabilities, Λ_1 of the star 1 and Λ_2 of the star 2, using the relation between Λ and M as given in figure 14 (a). The orange lines in figure 14 (b) refer to the constraints (solid:90% and dashed:50%) for which Λ_1 and Λ_2 are sampled independently [41], while the green lines refer to the constraints for which Λ_1 and Λ_2 are related through $\Lambda_a(\Lambda_s, q)$ with $\Lambda_a = (\Lambda_2 - \Lambda_1)/2$, $\Lambda_s = (\Lambda_2 + \Lambda_1)/2$, and $q = M_2/M_1$. In figure 14 (b) we clearly see that our predicted band is located within the 90% contours of the LIGO-Virgo data [41, 100].



(a)

(b)

Figure 14. (a) Tidal deformability Λ calculated from our EoS (b) Correlation of tidal deformabilities, Λ_1 and Λ_2; see the text for details.



(a)

(b)

Figure 15. (a) EoSs deduced from the observational M-R data of qLMXBs, thermonuclear bursters [2, 37, 38], and NICER data of PSR J0030+0451 [43]. The shaded blue and the hatched orange regions represent the 68% uncertainty band (evaluated in the bagging method) for the analyses with and without the NICER data, respectively. (b) M-R relations corresponding to the deduced EoSs shown in (a).

So far we have only used the observational data from the thermonuclear bursters and qLMXBs, which may contain large systematic errors related with the uncertain atmospheric model of neutron stars. The results in figures 15 (a) and (b) include the M-R constraint from the NICER mission as well. There are two independent analyses, as spotted on figure 15 (b), on the same observation of PSR J0030+0451 [43, 44], and we adopt the one [green bar in (b)] in ref. [43]. We see that the uncertainty becomes slightly larger by the inclusion of the NICER data, which is attributed to the relatively large deviation of the NICER data from others.

4.5 Possible EoSs with a weak first-order phase transition

In the analyses we have presented so far, we used the piecewise polytrope with 5 segments of density. Here, we change the number of segments from 5 to 7 and repeat the inference

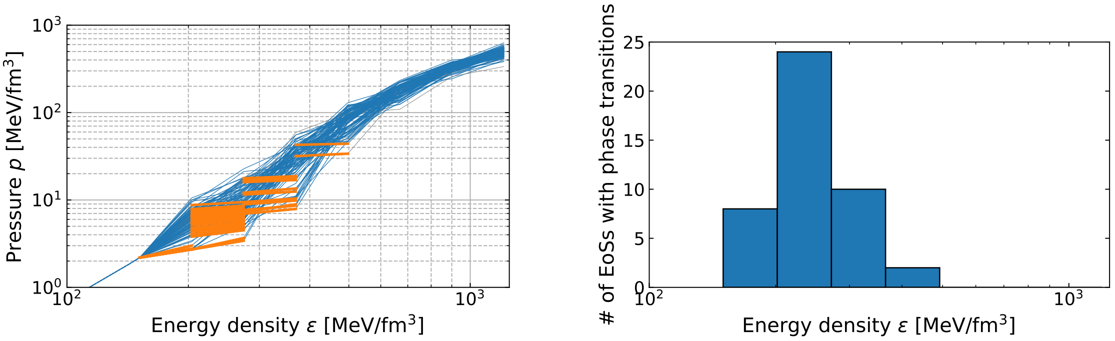
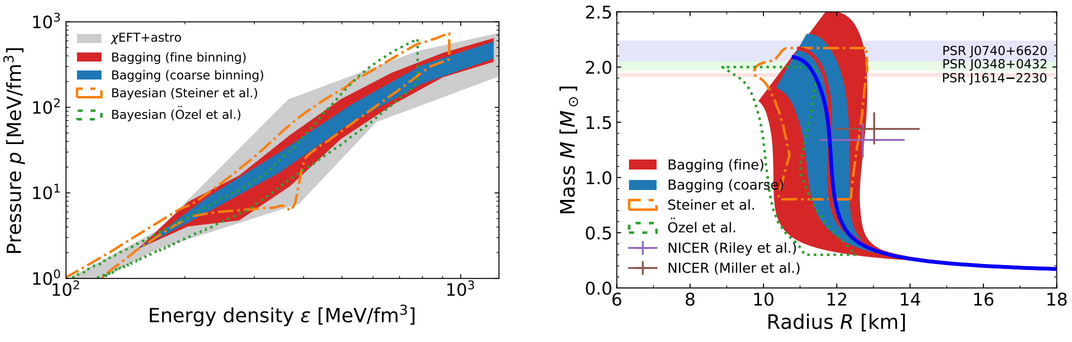


Figure 16. (Left) 100 output EoSs predicted from each bagging predictor with a first-order transition highlighted by the orange thick lines. (Right) Histogram of the first-order phase transitions in each energy density region of piecewise polytropes.

with the finer bins. There are mainly two issues argued in this subsection: a possibility of a weak first-order phase transition with the finer bins and its implication on the uncertainty quantification.

To this end we prepared $N = 100$ NNs in the bagging outlined in section 4.3. We note that each NN is trained so as to predict an EoS in response to the real observational data. Here we use the M-R data of qLMXBs and thermonuclear bursters without the NICER data. In the left panel of figure 16 we show 100 EoSs predicted from 100 independent NNs. We remind that the activation function in the output layer is chosen to be tanh which takes a value over $[-1, 1]$. When the predicted values is smaller than the threshold: $c_s^2 < \delta = 0.01$, we adjust it to $c_s^2 = \delta$ and identify a first-order phase transition. We found 44 predicted EoSs out of 100 that have a first-order phase transition. We highlight the region of first-order phase transition with orange thick lines in figure 16. From this plot we can understand why we increased the number of segments. If we use the EoS parametrization with 5 segments, weak first-order phase transitions are too strongly prohibited by coarse discretization.

We also make a histogram in the right panel of figure 16 to show a breakdown of the EoS regions with a first-order phase transition. This histogram counts the number of first-order transition EoSs in each energy density region. It is interesting to see that the most of the first-order phase transition is centered around the energy region 202–272 MeV. On the one hand, in the lower energy region 150–202 MeV the first-order phase transition is less likely, and this tendency is consistent with the fact that a stronger first-order phase transition in a lower energy region is more disfavored by the two-solar-mass pulsar constraint [101]. In the higher energy region, on the other hand, there are also less EoSs with a first-order phase transition. One may think that a first-order phase transition would be more allowed in the higher energy region, but it is not the case in the NN analysis. In section 3.4 we already discussed that the NN model tends to predict the most conservative value around $c_s^2 \sim 0.5$ in the high energy density regions where the constraints are inadequate. Therefore, the correct interpretation of the absence of the first-order transition in the high density regions as shown in figure 16 should be, not that our results exclude a first-order transition there,



(a) (b)

Figure 17. (a) EoSs deduced from the NN inference. The red and the blue shades represent the 68% credibility band evaluated in the bagging method with piecewise polytrope with 7 (fine bin) and 5 (coarse bin) segments, respectively. Other bands are the same as in figure 13. (b) M-R relations corresponding to the deduced EoSs and uncertainty bands in (a).

but merely that the observational data analyzed in our NN method does not favor a first-order transition there. Another artificial factor in the high energy density region is that our piecewise polytrope is equally spaced in the log scale, so the higher density segments have larger energy widths in the linear scale. Then, the parametrization does not have a resolution to represent a weak first-order phase transition in the high energy region. Here, we also make a comment that our NN approach does not take account of the third family scenario [102, 103] in which separate branches may appear in the M-R relation (see also refs. [66, 104] for recent discussions).

In figure 17 we show the EoS results with 7 segments (fine binning) after bagging. We see that uncertainty is widened compared with the results with 5 segments (coarse binning). This is partially because the number of output parameters, $c_{s,i}^2$ $(i = 1, \ldots, 7)$, is increased, while the amount of the input data is unchanged, so the corresponding uncertainty should increase. Furthermore we see that the uncertainty is enhanced by the effect of first-order phase transition. If there is a first-order phase transition, c_s^2 becomes (nearly) zero and these zero values enlarge the variance of the output in the bagging and increase the uncertainty accordingly. Now the uncertainty appears comparable with other Bayesian methods as seen in figure 17. Our previous results with 5 segments as shown in ref. [82] and in section 4.4 of this paper had smaller uncertainty, which implies that the choice of 5 segments turns out to be optimal.

Finally we plot the speed of sound in figure 18. The bare output from the NN model actually comprises these values of $c_{s,i}^2$. It is clear from the plots that the speed of sound exceeds the conformal limit $c_s^2 = 1/3$; see refs. [13, 105] for thorough discussions on the conformal limit. The inclusion of the NICER data only slightly widens uncertainty, and changing the segment number is a more dominant effect on uncertainty bands as quantified in the left panel of figure 18.

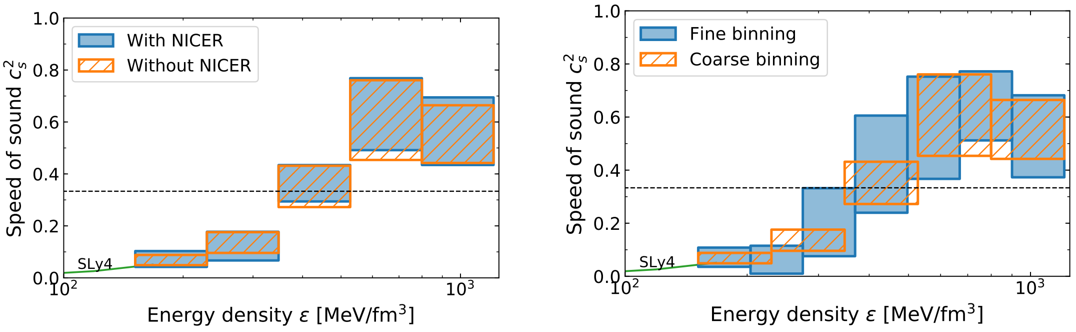


Figure 18. (Left) The speed of sound from our EoSs with and without the NICER data by the shaded blue and the hatched orange regions. (Right) The speed of sound from fine binning (shaded blue) and coarse binning (hatched orange) estimates.

5 More on the performance test: taming the overfitting

In section 3.2, we observed a quantitative difference between the learning curves for the training data sets with and without data augmentation by $n_s = 100$ as demonstrated in figure 5. Then, it would be a natural anticipation to consider that this observational data augmentation may be helpful to overcome the problems of local minimum trapping and overfitting that we often meet during the NN training. This section is aimed to discuss numerical experiments to understand the behavior of the learning curve and the role of n_s thereof. In particular, we will focus on the overfitting problem here.[4]

5.1 A toy model

Here we consider a specific and simple problem as a toy model. We choose the following double-peaked function $f(x)$:[5]

$$f(x) = (10x^2 - 6x^3 + x^4)e^{-x} \tag{5.1}$$

in the domain of $x \in [0, 10]$. A concrete shape of above $f(x)$ is depicted in figure 19. In table 3 we make a comparison between the neutron star calculation and this toy-model study of $f(x)$ fitting.

For the training data, we first generate a set of the "true" inputs, denoted by x^*, in the interval $[0, 10)$ and calculate the corresponding outputs, $y^* = f(x^*)$. We denote this base data set as $T^* = \{(x_p^*, y_p^*)\}_{p=1}^{n_{\text{base}}} \subset \mathbb{R}^2$ with $n_{\text{base}} = |T^*|$ being the size of the base data set. Then, similarly to the neutron star case, we augment the training data set by duplicating n_s copies of the base data with the input fluctuated by "observational uncertainty", Δx. Here, we set the distribution of Δx as the normal distribution with the standard deviation

[4] We also tested the performance to escape from the local minimum trapping, but the data augmentation seems not to be useful to resolve the trapping problem.

[5] This choice of $f(x)$ is motivated from spectral functions which often appear in the lattice-QCD calculation.

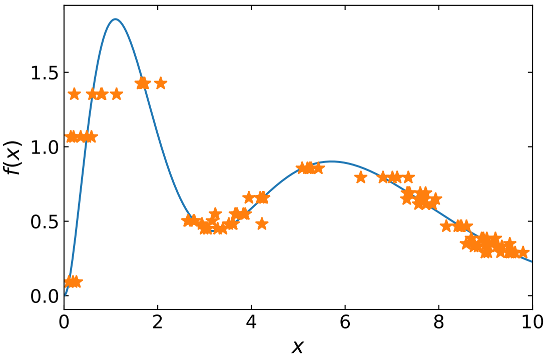


Figure 19. A concrete shape of $f(x)$ of our choice (blue curve) and typical training data (orange dots) for $n_{\text{base}} = 20$ and $n_s = 5$.

	Neutron star ($i = 1, \dots, 15$)	$f(x)$ fitting
Input	(M_i, R_i)	x
Output	EoS	$f(x)$
Training data (input)	$(M_i^* + \Delta M_i, R_i^* + \Delta R_i)$	$x^* + \Delta x$
Number of training data (output)	1948	$n_{\text{base}} = 20$
Number of training data (input)	1948 n_s = 1948000	$n_{\text{base}} \cdot n_s = 20\ n_s$

Table 3. Literal correspondence between the neutron star calculation (mock data analysis) and the $f(x)$ fitting.

fixed as $\sigma(x^*) = 0.3(x^* + 0.5)$ of our choice.[6] Then, the whole training data set can be expressed as

$$T = \left\{(x_p^* + \Delta x_{p,i}, y_p^*) \mid p \in \{1, \dots, n_{\text{base}}\}, i \in \{1, \dots, n_s\}, \Delta x_{p,i} \sim \mathcal{N}(0, \sigma^2(x_p^*))\right\}. \quad (5.2)$$

The size of the training data set T is found to be $|T| = n_s n_{\text{base}}$. The trivial case with $n_s = 1$ corresponds to the naive training data set without augmentation, while $n_s > 1$ augments the training data set. Figure 19 exemplifies the training data set with $n_{\text{base}} = 20$ and $n_s = 5$.

Since the task we deal with here is idealized and as simplified as possible, we can reasonably keep the NN architecture simple and relatively small (albeit deep). In the present study we basically use two types: `NN1221` and `NN1991` as tabulated in table 4. We also choose the simplest loss function here as `mae`, i.e., the mean absolute errors:

$$\ell_{\mathtt{abs}}(\boldsymbol{y}, \boldsymbol{y}') \equiv |\boldsymbol{y} - \boldsymbol{y}'| . \quad (5.3)$$

5.2 Input noises, parameter fluctuations, and dropout

To deepen our understanding of the data augmentation, we argue that including noises in the input is equivalent to introducing fluctuations to the NN parameters in the first layer.

[6] With this choice of $\sigma(x^*)$ the uncertainty is smaller in the x region with a narrow peak. For the demonstration purpose the toy model is designed in such a way that the problem is not too easy but not intractable.

Layer index	Neurons	Activation
0	1	N/A
1	2	Leaky ReLU
2	2	Leaky ReLU
3	1	Linear

(`NN1221`)

Layer index	Neurons	Activation
0	1	N/A
1	9	Leaky ReLU
2	9	Leaky ReLU
3	1	Linear

(`NN1991`)

Table 4. NN architectures used in this work: `NN1221` (left) and `NN1991` (right). We have chosen the leaky ReLU $f(x) = \alpha x$ $(x < 0)$, x $(x \geq 0)$ to avoid the dying ReLU problem. We fixed the leaky ReLU parameter $\alpha = 0.1$.

For the moment we will not limit ourselves to the special toy model in eq. (5.1) but here we shall develop a general formulation using the schematic notation introduced in section 2.4.

We consider the following case that a noise $\tilde{\boldsymbol{e}}$ is added to input values: $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \tilde{\boldsymbol{e}}$. The noise is assumed to be independent of the input value, i.e., $\Pr(\boldsymbol{x}, \tilde{\boldsymbol{e}}) = \Pr(\boldsymbol{x}) \Pr(\tilde{\boldsymbol{e}})$. In this case the input noise $\tilde{\boldsymbol{e}}$ can be absorbed by redefinition of the first layer parameters; that is,

$$\boldsymbol{x}^{(1)} = \sigma^{(1)}(W^{(1)}(\boldsymbol{x} + \tilde{\boldsymbol{e}}) + \boldsymbol{b}^{(1)}) = \sigma^{(1)}(W^{(1)}\boldsymbol{x} + \tilde{\boldsymbol{b}}^{(1)}) \,, \tag{5.4}$$

where $\tilde{\boldsymbol{b}}^{(1)} \equiv \boldsymbol{b}^{(1)} + W^{(1)}\tilde{\boldsymbol{e}}$. Therefore, we find,

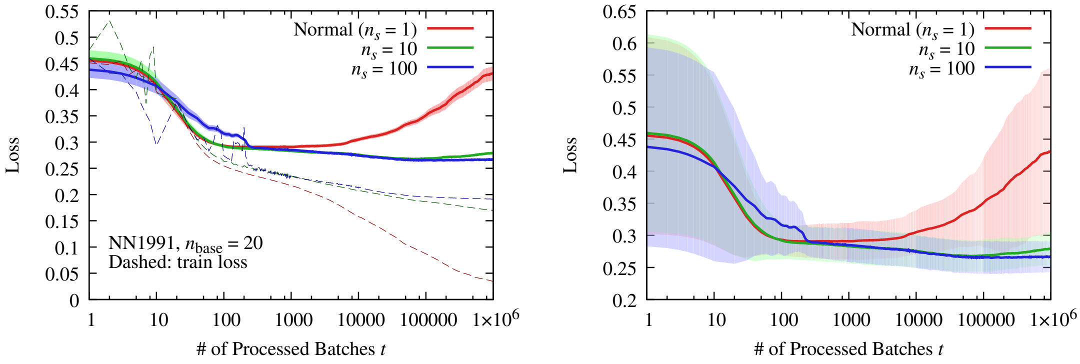
$$f(\tilde{\boldsymbol{x}}; W^{(1)}, \boldsymbol{b}^{(1)}, \ldots, W^{(L)}, \boldsymbol{b}^{(L)}) = f(\boldsymbol{x}; W^{(1)}, \tilde{\boldsymbol{b}}^{(1)}, \ldots, W^{(L)}, \boldsymbol{b}^{(L)}) \,. \tag{5.5}$$

The loss function with the input noise is given by

$$\begin{aligned} \mathcal{L}(\{W^{(\ell)}, \boldsymbol{b}^{(\ell)}\}_\ell) &= \int d\boldsymbol{x} d\tilde{\boldsymbol{e}} \Pr(\boldsymbol{x}, \tilde{\boldsymbol{e}}) \ell(\boldsymbol{y}, f(\tilde{\boldsymbol{x}}; W^{(1)}, \boldsymbol{b}^{(1)}, \ldots, W^{(L)}, \boldsymbol{b}^{(L)})) \\ &= \int d\tilde{\boldsymbol{e}} \Pr(\tilde{\boldsymbol{e}}) \int d\boldsymbol{x} \Pr(\boldsymbol{x}) \ell(\boldsymbol{y}, f(\boldsymbol{x}; W^{(1)}, \tilde{\boldsymbol{b}}^{(1)}, \ldots, W^{(L)}, \boldsymbol{b}^{(L)})) \\ &= \int d\tilde{\boldsymbol{e}} \Pr(\tilde{\boldsymbol{e}}) \mathcal{L}(W^{(1)}, \boldsymbol{b}^{(1)} + W^{(1)}\tilde{\boldsymbol{e}}, \ldots, W^{(L)}, \boldsymbol{b}^{(L)}) \,. \end{aligned} \tag{5.6}$$

This is nothing but the loss function with fluctuations in the first-layer biases $\boldsymbol{b}^{(1)}$; in summary, the noise $\tilde{\boldsymbol{e}}$ in the input is translated into the fluctuation by $W^{(1)}\tilde{\boldsymbol{e}}$ in the NN parameter $\boldsymbol{b}^{(1)}$. Once we realize this correspondence, we can generalize this latter prescription of fluctuating NN parameters not only in the first-layer $\boldsymbol{b}^{(1)}$ but also in all the parameters $\{W^{(\ell)}, \boldsymbol{b}^{(\ell)}\}_\ell$. We will test this idea of generalization later.

We can further relate the parameter fluctuations to a standard NN technique called *dropout*. Instead of adding random fluctuations to the parameters (*additive parameter noise*), we can also think of multiplying random fluctuations to the parameters (*multiplicative parameter noise*), i.e., $W_{ij}^{(\ell)} \to \tilde{W}_{ij}^{(\ell)} = W_{ij}^{(\ell)} \tilde{e}_{ij}^{(\ell)}$ and $b_i^{(\ell)} \to \tilde{b}_i^{(\ell)} = b_i^{(\ell)} \tilde{e}_i^{(\ell)}$ for a fixed ℓ, where $\tilde{e}_{ij}^{(\ell)}$ and $\tilde{e}_i^{(\ell)}$ are random fluctuations following a certain distribution. As a special case, let us assume that $\tilde{e}_{ij}^{(\ell)}$ and $\tilde{e}_i^{(\ell)}$ follow the simplest discrete probability distribution: the Bernoulli distribution $B(1, p)$, where 0 and 1 appear with probability p and $1 - p$, respectively. With an additional constraint of $\tilde{e}_{ij}^{(\ell)} = \tilde{e}_i^{(\ell)}$, putting these multiplicative parameter noise is exactly the procedure commonly referred to as "dropout" with the dropout rate identified as p.



(a) (b)

Figure 20. (a) Numerical experiment of increasing n_s. The mean values of the loss functions are represented by the dashed and the solid curves for the training and the validation losses, respectively. The bands represents the standard errors. (b) The same plot with the bands representing the standard deviations.

In this way we can understand the parameter noises as generalization of the data augmentation by noise injection and the dropout. It is already well known that the dropout is indeed an established tool to avoid the overfitting problem. Thus, we can expect that our observational data augmentation, which was originally needed to handle the observational uncertainties, have an effect similar to the dropout and can tame the overfitting behavior.

5.3 Numerical tests

Motivated by the relations between the data augmentation and the dropout through parameter noise, as argued in the previous subsection, we shall make a numerical comparison involving all of the data augmentation, the NN parameter noise, and the dropout using the toy model defined in eq. (5.1). For the parameter fluctuation noise, we consider the additive parameter noise of the normal distribution $\mathcal{N}(0, \sigma^2)$ in the first-layer parameters, $(W^{(1)}, \boldsymbol{b}^{(1)})$, or in all the parameters, $\{W^{(\ell)}, \boldsymbol{b}^{(\ell)}\}_\ell$. To manipulate this special type of the training presented in this section, we implement the NN and its optimization from scratch in `C++`. As an ideal test case we want a "troublesome" NN suffering from the overfitting problem. We have first verified that the NN method with `NN1221` in table 4 can reasonably solve the reconstruction problem of $f(x)$ while `NN1991` encounters the overfitting problem. In what follows below, we will present the results from `NN1991` to analyze the overfitting in details. We fix $n_{\text{base}} = 20$ and the mini-batch size is 10. We repeat carrying out the independent training procedures with the same setup 100 times and take the average of the loss function to draw the learning curves with probabilistic distribution.

In figure 20 we show the averaged learning curves for several choices of n_s to check the effect of the data augmentation. The mean values of the loss function $\langle \mathcal{L}(t) \rangle$ are represented by the dashed (training) and the solid (validation) curves along with the standard error band, $\Delta\langle \mathcal{L}(t) \rangle$, for the validation loss. Here the (population) mean is estimated by the sample mean over the 100 training, i.e., $\langle \mathcal{L}(t) \rangle \simeq \overline{\mathcal{L}(t)} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_i(t)$ with $N = 100$, where t is the training time in the unit of the net number of processed mini-batches, and

i is the index of the training. We should stress that this process with $N = 100$ is not the bagging, but we do so to quantify the performance on average. We also note that the learning curves of the validation loss fluctuate, and the standard deviation $\Delta\mathcal{L}(t)$ [as shown in figure 20 (b)] is much larger than the standard error band [as read from figure 20 (a)] that quantifies the typical deviation of the sample mean $\overline{\mathcal{L}(t)}$ from the "true" *population mean* $\langle\mathcal{L}(t)\rangle$. Here, the standard deviation and the standard error are estimated by the unbiased standard variance:[7]

$$\Delta\mathcal{L}(t) \equiv \sqrt{\overline{[\mathcal{L}(t) - \langle\mathcal{L}(t)\rangle]^2}} \simeq \sqrt{\frac{N}{N-1}\overline{[\mathcal{L}_i(t) - \langle\mathcal{L}(t)\rangle]^2}}\,, \tag{5.7}$$

$$\Delta\langle\mathcal{L}(t)\rangle \equiv \sqrt{\overline{[\overline{\mathcal{L}(t)} - \langle\mathcal{L}(t)\rangle]^2}} = \frac{1}{\sqrt{N}}\Delta\mathcal{L}(t)\,. \tag{5.8}$$

As we speculated before, the results in figure 20 (a) evidence that incorporating the data augmentation with $n_s > 1$ reduces the overfitting behavior (i.e., the validation loss increasing while the training loss decreasing). Obviously the learning curve for $n_s = 1$ indicates the overfitting. For a sufficiently large $n_s \gtrsim 10$, the validation loss does not increase any more and saturates towards a certain value. The improvement with $n_s \gtrsim 10$ can be understood from the size of the data set, $n_s n_{\text{base}} \gtrsim 10\cdot 20 = 200$, which is larger than the number of the NN parameters, i.e., 118 for `NN1991`. When n_s is small, the size of the data set is too small as compared to the NN parameters, so that the NN parameters cannot be well constrained. Even for $n_s = 100$ there are still discrepancies between the training and the validation losses as seen in figure 20 (a), which can be explained as follows; the size of the independent training base data with $n_{\text{base}} = 20$ is too small. The data augmentation does not supplement any additional information on the training data T but the contained information is the same as the base data set T^*. Therefore, the improvement should be eventually saturated with increasing n_s.

The standard deviation in figure 20 (b) at an early time is large reflecting the variance of initial NN parameters. The variance of the initial parameters quickly disappears around $t \lesssim 100$. An interesting observation is that the standard deviation becomes larger again when the overfitting occurs, and its magnitude is of the same order as the loss increase by the overfitting. This means that in some trials of training the performance does not necessarily get worse even in the overfitting time scale. The actual distribution of the final validation loss at $t = 10^6$ is shown in figure 21 in a form of histogram. We can see that the validation loss spreads to larger values in the overfitting case ($n_s = 1$), while the validation loss with the data augmentation ($n_s = 10$ and 100) is well localize around $\mathcal{L} = 0.25$.

In figure 22 we next check the effect of noise in the NN parameters. We introduce the additive parameter noise of the normal distribution $\mathcal{N}(0, \sigma^2)$ with several variances $\sigma = 0$, 0.2, and 0.4 to the first-layer parameters $(W^{(1)}, b^{(1)})$. From figure 22 we can confirm the same trend as the observational data augmentation, and indeed the overfitting is evaded similarly. We have tried different values of $\sigma = 0.2$ and 0.4, but they are consistent with

[7] The second line can be intuitively understood from the fact that the distribution of the mean $\langle\mathcal{L}(t)\rangle$ narrows with the variance $(\Delta\mathcal{L}(t))^2/N$ by the central limit theorem.

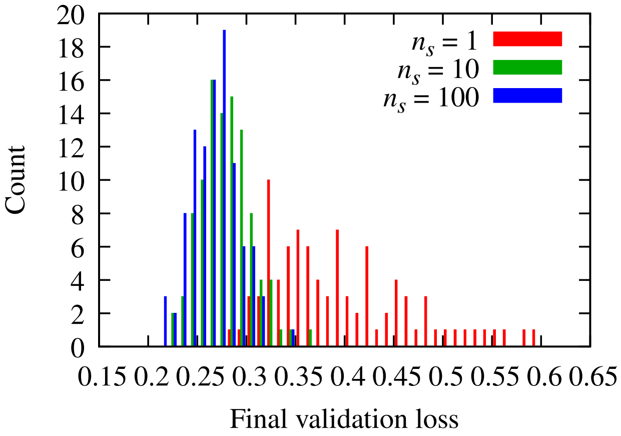


Figure 21. Histogram to show the 100 training distribution of the final validation loss at $t = 10^6$.

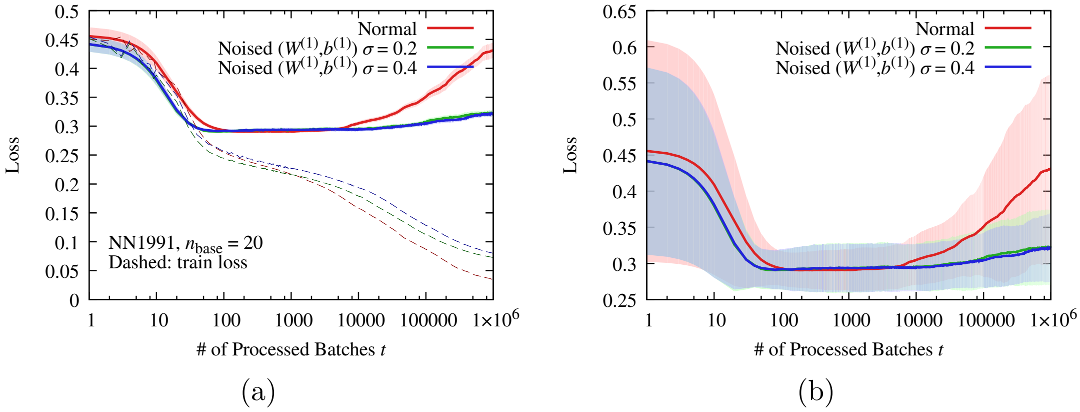


Figure 22. (a) Numerical experiment of adding noise to $(W^{(1)}, b^{(1)})$. (b) The same plot with the bands representing the standard deviations.

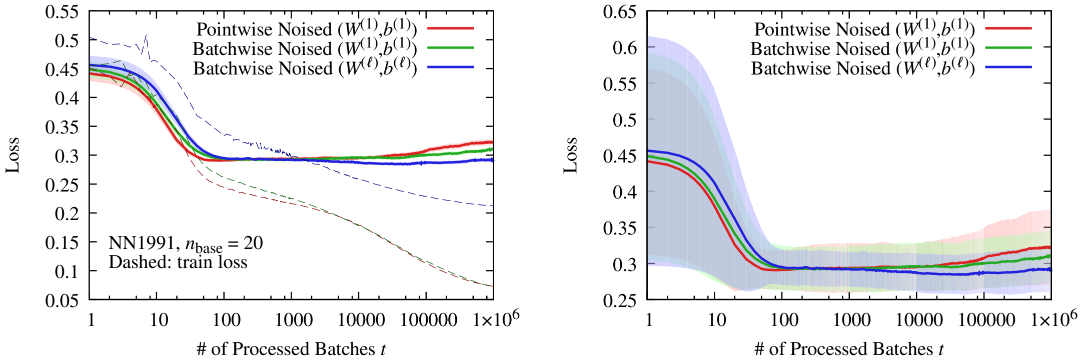
each other within the standard errors. We can also observe the behavior of the standard deviation similar to the case of the data augmentation.

We also compare different ways to introduce noise in the NN parameters. We here consider the following types of the parameter noise to make a plot in figure 23:

1. Add random values sampled from the normal distribution $\mathcal{N}(0, \sigma^2)$ with $\sigma = 0.2$ to the first-layer parameters $(W^{(1)}, b^{(1)})$. We resample parameter noises every time for each sample point in the training data ["Pointwise Noised $(W^{(1)}, b^{(1)})$" in figure 23].

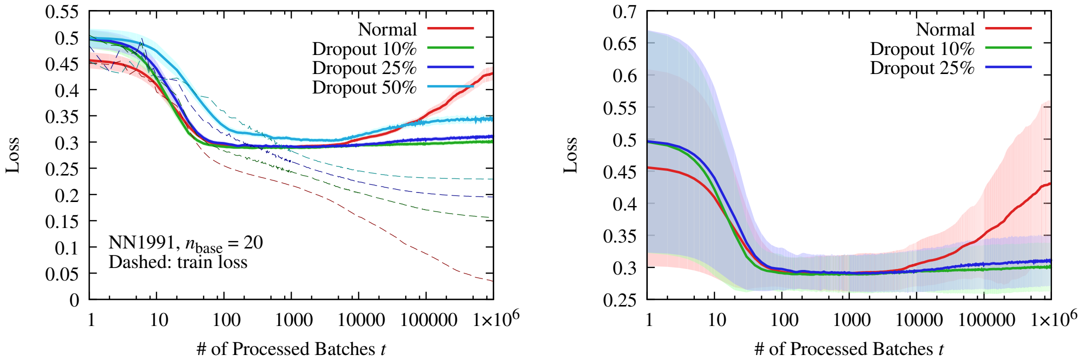
2. Add random values in the same way as (i) but resample parameter noises every time a mini-batch starts. We use a common set of the parameter noise values throughout a mini-batch. ["Batchwise Noised $(W^{(1)}, b^{(1)})$" in figure 23].

3. Add random values sampled from the normal distribution $\mathcal{N}(0, \sigma^2)$ with $\sigma = 0.2$ to all the NN parameters $(W^{(\ell)}, b^{(\ell)})$. We resample parameter noises every time a mini-batch starts as in (ii). ["Batchwise Noised $(W^{(\ell)}, b^{(\ell)})$" in figure 23].



(a) (b)

Figure 23. (a) Numerical experiment of adding noise in different ways; see the text for more details. (b) The same plot with the bands representing the standard deviations.



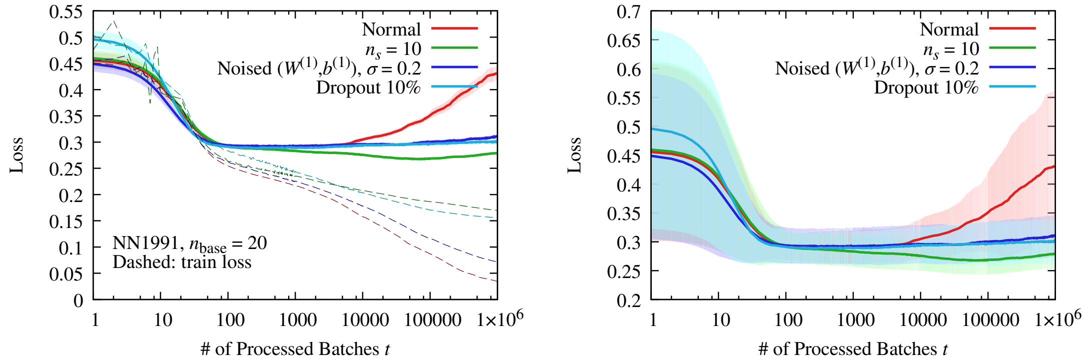
(a) (b)

Figure 24. (a) Numerical experiment of inserting a dropout. (b) The same plot with the bands representing the standard deviations.

We find that the performance is slightly improved for the batchwise noise rather than the pointwise noise. Also, better performance, i.e., smaller validation losses are reached for all-layer noises as prescribed in (iii) rather than for the first-layer noises as in (i) and (ii).

Finally, we plot the results for the case with a dropout in the NN in figure 24. Here again, we can validate the role of the dropout to overcome the overfitting problem. We have tried several different dropout rates; namely, 10%, 20%, and 50%, and numerically confirmed that the performance is better for small values of the dropout rate in the current setup.

Figure 25 is the summary plot of the performance comparison between three different approaches; namely, the observational data augmentation, the parameter noise, and the dropout. The parameter noise and the dropout results are close to each other and almost identical within the errors. Among these three approaches, it has turned out that the best performance is realized for the observational data augmentation in the present toy-model setup. The best strategy can be different for different model circumstances. The important point is not a question of which is the best but the finding that the data augmentation can



(a)

(b)

Figure 25. (a) Summary of the numerical experiment with the standard error bands. (b) The same plot with the bands representing the standard deviations.

tame the overfitting problem at the same level as the standard procedure of the dropout. This is a very interesting observation. In physics all observational data has uncertainties, and the data is intrinsically augmented by repeated measurements with limited accuracy. Then, for physics problems, this NN nature of the "observational data augmentation" can naturally evade the overfitting problem by itself.

6 Summary and outlooks

In this work, we have performed a comprehensive analysis of inferring the equation of state (EoS) from the observed masses and radii of neutron stars using the previously proposed method of deep machine learning [81, 82]. Following the method elaborated in section 2, we have subdivided our analyses into three parts: the examination of the deep learning methodology using the mock data mimicking the real observations (section 3), the application of the method to the actual observational data of neutron stars (section 4), and the discussion of our *observational data augmentation* by noise fluctuations in a simple setup idealized for theorization (section 5).

In the first part (section 3) we have demonstrated how our method works by presenting the typical two examples of the EoS prediction. Then we have quantified the uncertainty or error between the prediction and the original value as given in eq. (3.1) and inspected the prediction reliability for the whole validation data set. The upshot is that, by looking at the histogram and the correlation between the prediction uncertainties in different energy density regions, we have verified that the EoS reconstruction fares well for the lower density regions, but the higher density regions are not sufficiently constrained. We have also compared the NN method to a more conventional approach, i.e., the polynomial regression. We then quantified the results in terms of $\Delta R_{\rm RMS}$, ΔR_σ, and $\Delta R_{2\sigma}$ as given in eqs. (3.4) and (3.5). We have reached a conclusion that our NN method leads to more robust outputs, and the polynomial regression may have outliers that damage the overall performance.

In the second part (section 4) we have applied the EoS inference for real neutron star data from the various sources; namely, qLMXBs, thermonuclear bursters, and the pulsar with hot spots on the surface, at which the NICER mission aims. We have also introduced natural and intuitive ways for uncertainty quantification. One is based on the validation loss and the other is based on the procedure called the bagging. We trained multiple NNs independently and took the average for the bagging. We have utilized independent outputs from the multiple NNs to discuss the likelihood of EoSs with a first-order phase transition. We made a histogram to count the rate of EoSs with a first-order transition in each energy density segment.

In the third part (section 5) we have brought to light a byproduct from our prescription of incorporating observational uncertainty. The training data is augmented with fluctuations corresponding to observational uncertainty, which we call the observational data augmentation. Then, the data augmentation with n_s fluctuating copies has turned out to tame the overfitting problem. We have explicitly confirmed this behavior of the overfitting reduction by monitoring a time evolution of the validation loss.

Although we have carried out extensive analyses in this work, there are still a lot to be done in the future to improve the applicability of our method to the neutron star EoS inference. As our NN method reliably predicts the most likely EoS in lower density region, it would be supplementary to provide information on the TOV limiting mass (the heaviest observed mass) directly in the NN input. At present we have incorporated this information indirectly by just kicking out the training data that does not reach the TOV limit. Also, our NN model currently copes with the pairs of mass-radius data only, but it would be an important next step to extend the NN model so as to process other forms of data, e.g., the tidal deformability.

Currently we employ the feedforward network. It is also a natural future extension to upgrade the NN architecture. For instance, we can replace the NN with the convolutional neural network (CNN) as frequently used in the image processing. With CNN we can make full use of the two-dimensional Bayesian posterior distribution for the input observable data. On top of the improvement in the NN part, we can also think of another extension of better EoS parametrization. It would be possible to use parametrizations other than the piecewise polytropes, such as the spectral representation [106, 107] and the quarkyonic parametrization [56]. In this work we jointed our predicted EoS above n_0 with the SLy4 below n_0. The dependence of the joint density and/or choices of the nuclear EoS should be systematically examined. Our naive expectation is that none of these improvements would affect the results drastically.

The uncertainty quantification is another interesting issue. In this work we have used the practical but oversimplified techniques; namely, the bagging and the validation loss. More rigorous treatments of uncertainties may be possible by adopting the CNN as mentioned above, or by combining the Bayesian method and NN approach. Using such a Bayesian NN, in which the network weights are sampled from a probability distribution, we can obtain the posterior probabilities of the outputs and thus we can capture the NN model uncertainties correctly. One efficient way to implement this computation is to set up a dropout as outlined in refs. [108, 109] (see also ref. [110] for physics applications). In analogy to the explanation in section 5.2, we can regard a dropout as a Bayesian inference with likelihood functions set to the Bernoulli distribution $B(1, p)$. All above-mentioned extensions deserve further investigations.

Acknowledgments

This work was was supported by JSPS KAKENHI Grant No. 20J10506 (YF) and Grant Nos. 18H01211, 19K21874 (KF).

References

[1] J.M. Lattimer, *The nuclear equation of state and neutron star masses*, *Ann. Rev. Nucl. Part. Sci.* **62** (2012) 485 [arXiv:1305.3510] [INSPIRE].

[2] F. Özel and P. Fréire, *Masses, Radii, and the Equation of State of Neutron Stars*, *Ann. Rev. Astron. Astrophys.* **54** (2016) 401 [arXiv:1603.02698] [INSPIRE].

[3] G. Baym, T. Hatsuda, T. Kojo, P.D. Powell, Y. Song and T. Takatsuka, *From hadrons to quarks in neutron stars: a review*, *Rept. Prog. Phys.* **81** (2018) 056902 [arXiv:1707.04966] [INSPIRE].

[4] L. Baiotti, *Gravitational waves from neutron star mergers and their relation to the nuclear equation of state*, *Prog. Part. Nucl. Phys.* **109** (2019) 103714 [arXiv:1907.08534] [INSPIRE].

[5] T. Kojo, *QCD equations of state and speed of sound in neutron stars*, 11, 2020 [arXiv:2011.10940] [INSPIRE].

[6] K. Hebeler and A. Schwenk, *Chiral three-nucleon forces and neutron matter*, *Phys. Rev. C* **82** (2010) 014314 [arXiv:0911.0483] [INSPIRE].

[7] S. Gandolfi, J. Carlson and S. Reddy, *The maximum mass and radius of neutron stars and the nuclear symmetry energy*, *Phys. Rev. C* **85** (2012) 032801 [arXiv:1101.1921] [INSPIRE].

[8] I. Tews, T. Krüger, K. Hebeler and A. Schwenk, *Neutron matter at next-to-next-to-next-to-leading order in chiral effective field theory*, *Phys. Rev. Lett.* **110** (2013) 032504 [arXiv:1206.0025] [INSPIRE].

[9] J.W. Holt, N. Kaiser and W. Weise, *Nuclear chiral dynamics and thermodynamics*, *Prog. Part. Nucl. Phys.* **73** (2013) 35 [arXiv:1304.6350] [INSPIRE].

[10] G. Hagen et al., *Coupled-cluster calculations of nucleonic matter*, *Phys. Rev. C* **89** (2014) 014319 [arXiv:1311.2925] [INSPIRE].

[11] A. Roggero, A. Mukherjee and F. Pederiva, *Quantum Monte Carlo calculations of neutron matter with non-local chiral interactions*, *Phys. Rev. Lett.* **112** (2014) 221103 [arXiv:1402.1576] [INSPIRE].

[12] G. Wlazłowski, J.W. Holt, S. Moroz, A. Bulgac and K.J. Roche, *Auxiliary-Field Quantum Monte Carlo Simulations of Neutron Matter in Chiral Effective Field Theory*, *Phys. Rev. Lett.* **113** (2014) 182503 [arXiv:1403.3753] [INSPIRE].

[13] I. Tews, J. Carlson, S. Gandolfi and S. Reddy, *Constraining the speed of sound inside neutron stars with chiral effective field theory interactions and observations*, *Astrophys. J.* **860** (2018) 149 [arXiv:1801.01923] [INSPIRE].

[14] C. Drischler, R.J. Furnstahl, J.A. Melendez and D.R. Phillips, *How Well Do We Know the Neutron-Matter Equation of State at the Densities Inside Neutron Stars? A Bayesian Approach with Correlated Uncertainties*, Phys. Rev. Lett. **125** (2020) 202702 [arXiv:2004.07232] [INSPIRE].

[15] C. Drischler, J.W. Holt and C. Wellenhofer, *Chiral Effective Field Theory and the High-Density Nuclear Equation of State*, arXiv:2101.01709 [INSPIRE].

[16] B.A. Freedman and L.D. McLerran, *Fermions and Gauge Vector Mesons at Finite Temperature and Density. 1. Formal Techniques*, Phys. Rev. D **16** (1977) 1130 [INSPIRE].

[17] B.A. Freedman and L.D. McLerran, *Fermions and Gauge Vector Mesons at Finite Temperature and Density. 3. The Ground State Energy of a Relativistic Quark Gas*, Phys. Rev. D **16** (1977) 1169 [INSPIRE].

[18] V. Baluni, *Nonabelian Gauge Theories of Fermi Systems: Chromotheory of Highly Condensed Matter*, Phys. Rev. D **17** (1978) 2092 [INSPIRE].

[19] A. Kurkela, P. Romatschke and A. Vuorinen, *Cold Quark Matter*, Phys. Rev. D **81** (2010) 105021 [arXiv:0912.1856] [INSPIRE].

[20] E.S. Fraga, A. Kurkela and A. Vuorinen, *Interacting quark matter equation of state for compact stars*, Astrophys. J. Lett. **781** (2014) L25 [arXiv:1311.5154] [INSPIRE].

[21] T. Gorda, A. Kurkela, P. Romatschke, M. Säppi and A. Vuorinen, *Next-to-Next-to-Next-to-Leading Order Pressure of Cold Quark Matter: Leading Logarithm*, Phys. Rev. Lett. **121** (2018) 202701 [arXiv:1807.04120] [INSPIRE].

[22] J. Ghiglieri, A. Kurkela, M. Strickland and A. Vuorinen, *Perturbative Thermal QCD: Formalism and Applications*, Phys. Rept. **880** (2020) 1 [arXiv:2002.10188] [INSPIRE].

[23] Y. Fujimoto and K. Fukushima, *Equation of state of cold and dense QCD matter in resummed perturbation theory*, arXiv:2011.10891 [INSPIRE].

[24] G. Aarts, *Introductory lectures on lattice QCD at nonzero baryon number*, J. Phys. Conf. Ser. **706** (2016) 022004 [arXiv:1512.05145] [INSPIRE].

[25] A. Akmal, V.R. Pandharipande and D.G. Ravenhall, *The Equation of state of nucleon matter and neutron star structure*, Phys. Rev. C **58** (1998) 1804 [nucl-th/9804027] [INSPIRE].

[26] H. Togashi, K. Nakazato, Y. Takehara, S. Yamamuro, H. Suzuki and M. Takano, *Nuclear equation of state for core-collapse supernova simulations with realistic nuclear forces*, Nucl. Phys. A **961** (2017) 78 [arXiv:1702.05324] [INSPIRE].

[27] F. Douchin and P. Haensel, *A unified equation of state of dense matter and neutron star structure*, Astron. Astrophys. **380** (2001) 151 [astro-ph/0111092] [INSPIRE].

[28] B.D. Serot and J.D. Walecka, *Recent progress in quantum hadrodynamics*, Int. J. Mod. Phys. E **6** (1997) 515 [nucl-th/9701058] [INSPIRE].

[29] M. Drews and W. Weise, *Functional renormalization group studies of nuclear and neutron matter*, Prog. Part. Nucl. Phys. **93** (2017) 69 [arXiv:1610.07568] [INSPIRE].

[30] P. Demorest, T. Pennucci, S. Ransom, M. Roberts and J. Hessels, *Shapiro Delay Measurement of A Two Solar Mass Neutron Star*, Nature **467** (2010) 1081 [arXiv:1010.5788] [INSPIRE].

[31] E. Fonseca et al., *The NANOGrav Nine-year Data Set: Mass and Geometric Measurements of Binary Millisecond Pulsars*, *Astrophys. J.* **832** (2016) 167 [arXiv:1603.00545] [INSPIRE].

[32] J. Antoniadis et al., *A Massive Pulsar in a Compact Relativistic Binary*, *Science* **340** (2013) 6131 [arXiv:1304.6875] [INSPIRE].

[33] NANOGrav collaboration, *Relativistic Shapiro delay measurements of an extremely massive millisecond pulsar*, *Nature Astron.* **4** (2019) 72 [arXiv:1904.06759] [INSPIRE].

[34] F. Ozel, G. Baym and T. Guver, *Astrophysical Measurement of the Equation of State of Neutron Star Matter*, *Phys. Rev. D* **82** (2010) 101301 [arXiv:1002.3153] [INSPIRE].

[35] A.W. Steiner, J.M. Lattimer and E.F. Brown, *The Equation of State from Observed Masses and Radii of Neutron Stars*, *Astrophys. J.* **722** (2010) 33 [arXiv:1005.0811] [INSPIRE].

[36] A.W. Steiner, J.M. Lattimer and E.F. Brown, *The Neutron Star Mass-Radius Relation and the Equation of State of Dense Matter*, *Astrophys. J. Lett.* **765** (2013) L5 [arXiv:1205.6871] [INSPIRE].

[37] F. Ozel, D. Psaltis, T. Guver, G. Baym, C. Heinke and S. Guillot, *The Dense Matter Equation of State from Neutron Star Radius and Mass Measurements*, *Astrophys. J.* **820** (2016) 28 [arXiv:1505.05155] [INSPIRE].

[38] S. Bogdanov, C.O. Heinke, F. Özel and T. Güver, *Neutron Star Mass-Radius Constraints of the Quiescent Low-mass X-ray Binaries X7 and X5 in the Globular Cluster 47 Tuc*, *Astrophys. J.* **831** (2016) 184 [arXiv:1603.01630] [INSPIRE].

[39] M.C. Miller, *Astrophysical Constraints on Dense Matter in Neutron Stars*, *Astrophys. Space Sci. Libr.* **461** (2020) 1 [arXiv:1312.0029] [INSPIRE].

[40] M.C. Miller and F.K. Lamb, *Observational Constraints on Neutron Star Masses and Radii*, *Eur. Phys. J. A* **52** (2016) 63 [arXiv:1604.03894] [INSPIRE].

[41] LIGO Scientific and Virgo collaborations, *GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral*, *Phys. Rev. Lett.* **119** (2017) 161101 [arXiv:1710.05832] [INSPIRE].

[42] LIGO Scientific and Virgo collaborations, *GW190425: Observation of a Compact Binary Coalescence with Total Mass* $\sim 3.4M_\odot$, *Astrophys. J. Lett.* **892** (2020) L3 [arXiv:2001.01761] [INSPIRE].

[43] T.E. Riley et al., *A NICER View of PSR J0030+0451: Millisecond Pulsar Parameter Estimation*, *Astrophys. J. Lett.* **887** (2019) L21 [arXiv:1912.05702] [INSPIRE].

[44] M.C. Miller et al., *PSR J0030+0451 Mass and Radius from NICER Data and Implications for the Properties of Neutron Star Matter*, *Astrophys. J. Lett.* **887** (2019) L24 [arXiv:1912.05705] [INSPIRE].

[45] K. Yagi and N. Yunes, *I-Love-Q*, *Science* **341** (2013) 365 [arXiv:1302.4499] [INSPIRE].

[46] K. Yagi and N. Yunes, *I-Love-Q Relations in Neutron Stars and their Applications to Astrophysics, Gravitational Waves and Fundamental Physics*, *Phys. Rev. D* **88** (2013) 023009 [arXiv:1303.1528] [INSPIRE].

[47] K. Yagi and N. Yunes, *Approximate Universal Relations for Neutron Stars and Quark Stars*, *Phys. Rept.* **681** (2017) 1 [arXiv:1608.02582] [INSPIRE].

[48] E. Annala, T. Gorda, A. Kurkela, J. Nättilä and A. Vuorinen, *Evidence for quark-matter cores in massive neutron stars*, *Nature Phys.* **16** (2020) 907 [arXiv:1903.09121] [INSPIRE].

[49] K. Masuda, T. Hatsuda and T. Takatsuka, *Hadron-Quark Crossover and Massive Hybrid Stars with Strangeness*, Astrophys. J. **764** (2013) 12 [arXiv:1205.3621] [INSPIRE].

[50] Y. Fujimoto, K. Fukushima and W. Weise, *Continuity from neutron matter to two-flavor quark matter with $^1S_0$ and $^3P_2$ superfluidity*, Phys. Rev. D **101** (2020) 094009 [arXiv:1908.09360] [INSPIRE].

[51] L. McLerran and R.D. Pisarski, *Phases of cold, dense quarks at large N_c*, Nucl. Phys. A **796** (2007) 83 [arXiv:0706.2191] [INSPIRE].

[52] K. Fukushima and T. Kojo, *The Quarkyonic Star*, Astrophys. J. **817** (2016) 180 [arXiv:1509.00356] [INSPIRE].

[53] L. McLerran and S. Reddy, *Quarkyonic Matter and Neutron Stars*, Phys. Rev. Lett. **122** (2019) 122701 [arXiv:1811.12503] [INSPIRE].

[54] K.S. Jeong, L. McLerran and S. Sen, *Dynamically generated momentum space shell structure of quarkyonic matter via an excluded volume model*, Phys. Rev. C **101** (2020) 035201 [arXiv:1908.04799] [INSPIRE].

[55] D.C. Duarte, S. Hernandez-Ortiz and K.S. Jeong, *Excluded-volume model for quarkyonic Matter: Three-flavor baryon-quark Mixture*, Phys. Rev. C **102** (2020) 025203 [arXiv:2003.02362] [INSPIRE].

[56] T. Zhao and J.M. Lattimer, *Quarkyonic Matter Equation of State in Beta-Equilibrium*, Phys. Rev. D **102** (2020) 023021 [arXiv:2004.08293] [INSPIRE].

[57] K. Fukushima, T. Kojo and W. Weise, *Hard-core deconfinement and soft-surface delocalization from nuclear to quark matter*, Phys. Rev. D **102** (2020) 096017 [arXiv:2008.08436] [INSPIRE].

[58] E.R. Most et al., *Signatures of quark-hadron phase transitions in general-relativistic neutron-star mergers*, Phys. Rev. Lett. **122** (2019) 061101 [arXiv:1807.03684] [INSPIRE].

[59] A. Bauswein et al., *Identifying a first-order phase transition in neutron star mergers through gravitational waves*, Phys. Rev. Lett. **122** (2019) 061102 [arXiv:1809.01116] [INSPIRE].

[60] E.R. Most, L. Jens Papenfort, V. Dexheimer, M. Hanauske, H. Stoecker and L. Rezzolla, *On the deconfinement phase transition in neutron-star mergers*, Eur. Phys. J. A **56** (2020) 59 [arXiv:1910.13893] [INSPIRE].

[61] D. Alvarez-Castillo, A. Ayriyan, S. Benic, D. Blaschke, H. Grigorian and S. Typel, *New class of hybrid EoS and Bayesian M-R data analysis*, Eur. Phys. J. A **52** (2016) 69 [arXiv:1603.03457] [INSPIRE].

[62] C.A. Raithel, F. Ozel and D. Psaltis, *From Neutron Star Observables to the Equation of State: An Optimal Parametrization*, Astrophys. J. **831** (2016) 44 [arXiv:1605.03591] [INSPIRE].

[63] C.A. Raithel, F. Özel and D. Psaltis, *From Neutron Star Observables to the Equation of State. II. Bayesian Inference of Equation of State Pressures*, Astrophys. J. **844** (2017) 156 [arXiv:1704.00737] [INSPIRE].

[64] G. Raaijmakers et al., *Constraining the dense matter equation of state with joint analysis of NICER and LIGO/Virgo measurements*, Astrophys. J. Lett. **893** (2020) L21 [arXiv:1912.11031] [INSPIRE].

[65] G. Raaijmakers et al., *A NICER view of PSR J0030+0451: Implications for the dense matter equation of state*, Astrophys. J. Lett. **887** (2019) L22 [arXiv:1912.05703] [INSPIRE].

[66] D. Blaschke, A. Ayriyan, D.E. Alvarez-Castillo and H. Grigorian, *Was GW170817 a Canonical Neutron Star Merger? Bayesian Analysis with a Third Family of Compact Stars*, Universe **6** (2020) 81 [arXiv:2005.02759] [INSPIRE].

[67] P. Landry and R. Essick, *Nonparametric inference of the neutron star equation of state from gravitational wave observations*, Phys. Rev. D **99** (2019) 084049 [arXiv:1811.12529] [INSPIRE].

[68] R. Essick, P. Landry and D.E. Holz, *Nonparametric Inference of Neutron Star Composition, Equation of State, and Maximum Mass with GW170817*, Phys. Rev. D **101** (2020) 063007 [arXiv:1910.09740] [INSPIRE].

[69] R. Essick, I. Tews, P. Landry, S. Reddy and D.E. Holz, *Direct Astrophysical Tests of Chiral Effective Field Theory at Supranuclear Densities*, Phys. Rev. C **102** (2020) 055803 [arXiv:2004.07744] [INSPIRE].

[70] L.-G. Pang, K. Zhou, N. Su, H. Petersen, H. Stöcker and X.-N. Wang, *An equation-of-state-meter of quantum chromodynamics transition from deep learning*, Nature Commun. **9** (2018) 210 [arXiv:1612.04262] [INSPIRE].

[71] Y. Mori, K. Kashiwa and A. Ohnishi, *Toward solving the sign problem with path optimization method*, Phys. Rev. D **96** (2017) 111501 [arXiv:1705.05605] [INSPIRE].

[72] Y.D. Hezaveh, L. Perreault Levasseur and P.J. Marshall, *Fast Automated Analysis of Strong Gravitational Lenses with Convolutional Neural Networks*, Nature **548** (2017) 555 [arXiv:1708.08842] [INSPIRE].

[73] S. Chang, T. Cohen and B. Ostdiek, *What is the Machine Learning?*, Phys. Rev. D **97** (2018) 056009 [arXiv:1709.10106] [INSPIRE].

[74] Z.M. Niu and H.Z. Liang, *Nuclear mass predictions based on Bayesian neural network approach with pairing and shell effects*, Phys. Lett. B **778** (2018) 48 [arXiv:1801.04411] [INSPIRE].

[75] B. Kaspschak and U.-G. Meißner, *How machine learning conquers the unitary limit*, Commun. Theor. Phys. **73** (2021) 035101 [arXiv:2003.09137] [INSPIRE].

[76] L. Wang, Y. Jiang, L. He and K. Zhou, *Continuous-mixture Autoregressive Networks for efficient variational calculation of many-body systems*, arXiv:2005.04857 [INSPIRE].

[77] D. George and E.A. Huerta, *Deep Neural Networks to Enable Real-time Multimessenger Astrophysics*, Phys. Rev. D **97** (2018) 044039 [arXiv:1701.00008] [INSPIRE].

[78] M. Carrillo, M. Gracia-Linares, J.A. González and F.S. Guzmán, *Parameter estimates in binary black hole collisions using neural networks*, Gen. Rel. Grav. **48** (2016) 141 [arXiv:1608.02491] [INSPIRE].

[79] D. George and E.A. Huerta, *Deep Learning for Real-time Gravitational Wave Detection and Parameter Estimation: Results with Advanced LIGO Data*, Phys. Lett. B **778** (2018) 64 [arXiv:1711.03121] [INSPIRE].

[80] D. George, H. Shen and E.A. Huerta, *Classification and unsupervised clustering of LIGO data with Deep Transfer Learning*, Phys. Rev. D **97** (2018) 101501 [INSPIRE].

[81] Y. Fujimoto, K. Fukushima and K. Murase, *Methodology study of machine learning for the neutron star equation of state*, Phys. Rev. D **98** (2018) 023019 [arXiv:1711.06748] [INSPIRE].

[82] Y. Fujimoto, K. Fukushima and K. Murase, *Mapping neutron star data to the equation of state using the deep neural network*, Phys. Rev. D **101** (2020) 054016 [arXiv:1903.03400] [INSPIRE].

[83] M. Ferreira and C. Providência, *Unveiling the nuclear matter EoS from neutron star properties: a supervised machine learning approach*, arXiv:1910.05554 [INSPIRE].

[84] F. Morawski and M. Bejger, *Neural network reconstruction of the dense matter equation of state derived from the parameters of neutron stars*, Astron. Astrophys. **642** (2020) A78 [arXiv:2006.07194] [INSPIRE].

[85] S. Traversi and P. Char, *Structure of Quark Star: A Comparative Analysis of Bayesian Inference and Neural Network Based Modeling*, Astrophys. J. **905** (2020) 9 [arXiv:2007.10239] [INSPIRE].

[86] R.C. Tolman, *Static solutions of Einstein's field equations for spheres of fluid*, Phys. Rev. **55** (1939) 364 [INSPIRE].

[87] J.R. Oppenheimer and G.M. Volkoff, *On Massive neutron cores*, Phys. Rev. **55** (1939) 374 [INSPIRE].

[88] L. Lindblom, *Determining the nuclear equation of state from neutron-star masses and radii*, Astrophys. J. **398** (1992) 569.

[89] G. Cybenko, *Approximation by superpositions of a sigmoidal function*, Math. Control Signals Syst. **2** (1989) 303.

[90] K. Hornik, *Approximation capabilities of multilayer feedforward networks*, Neural Netw. **4** (1991) 251.

[91] F. Chollet, *Keras: Deep learning library for theano and tensorflow*, https://github.com/fchollet/keras (2015).

[92] M. Abadi et al., *Tensorflow: A system for large-scale machine learning*, arXiv:1605.08695.

[93] D.P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980 [INSPIRE].

[94] C. Zhang, S. Bengio, M. Hardt, B. Recht and O. Vinyals, *Understanding deep learning requires rethinking generalization*, published in *5th International Conference on Learning Representations (ICLR 2017)*, Toulon France (2017) [arXiv:1611.03530].

[95] Z. Allen-Zhu, Y. Li and Y. Liang, *Learning and generalization in overparameterized neural networks, going beyond two layers*, in *Advances in Neural Information Processing Systems. Vol. 32 (NeurIPS 2019)*, Vancouver Canada (2019), H. Wallach et al. eds., Curran Associates, Inc., Red Hook U.S.A. (2019), pg. 6158, https://proceedings.neurips.cc/paper/2019/file/62dad6e273d32235ae02b7d321578ee8-Paper.pdf.

[96] L. Breiman, *Bagging predictors*, Mach. Learn. **24** (1996) 123.

[97] K. Hebeler, J.M. Lattimer, C.J. Pethick and A. Schwenk, *Equation of state and neutron star properties constrained by nuclear physics and observation*, Astrophys. J. **773** (2013) 11 [arXiv:1303.4662] [INSPIRE].

[98] T. Hinderer, *Tidal Love numbers of neutron stars*, *Astrophys. J.* **677** (2008) 1216 [arXiv:0711.2420] [INSPIRE].

[99] T. Hinderer, B.D. Lackey, R.N. Lang and J.S. Read, *Tidal deformability of neutron stars with realistic equations of state and their gravitational wave signatures in binary inspiral*, *Phys. Rev. D* **81** (2010) 123016 [arXiv:0911.3535] [INSPIRE].

[100] LIGO Scientific, Virgo collaboration, *GW170817: Measurements of neutron star radii and equation of state*, *Phys. Rev. Lett.* **121** (2018) 161101 [arXiv:1805.11581] [INSPIRE].

[101] M.G. Alford, G.F. Burgio, S. Han, G. Taranto and D. Zappalà, *Constraining and applying a generic high-density equation of state*, *Phys. Rev. D* **92** (2015) 083002 [arXiv:1501.07902] [INSPIRE].

[102] U.H. Gerlach, *Equation of State at Supranuclear Densities and the Existence of a Third Family of Superdense Stars*, *Phys. Rev.* **172** (1968) 1325 [INSPIRE].

[103] K. Schertler, C. Greiner, J. Schaffner-Bielich and M.H. Thoma, *Quark phases in neutron stars and a 'third family' of compact stars as a signature for phase transitions*, *Nucl. Phys. A* **677** (2000) 463 [astro-ph/0001467] [INSPIRE].

[104] M.G. Alford and A. Sedrakian, *Compact stars with sequential QCD phase transitions*, *Phys. Rev. Lett.* **119** (2017) 161104 [arXiv:1706.01592] [INSPIRE].

[105] P. Bedaque and A.W. Steiner, *Sound velocity bound and neutron stars*, *Phys. Rev. Lett.* **114** (2015) 031103 [arXiv:1408.5116] [INSPIRE].

[106] L. Lindblom, *Spectral Representations of Neutron-Star Equations of State*, *Phys. Rev. D* **82** (2010) 103011 [arXiv:1009.0738] [INSPIRE].

[107] L. Lindblom and N.M. Indik, *Spectral Approach to the Relativistic Inverse Stellar Structure Problem II*, *Phys. Rev. D* **89** (2014) 064003 [*Erratum ibid.* **93** (2016) 129903] [arXiv:1310.0803] [INSPIRE].

[108] Y. Gal and Z. Ghahramani, *Dropout as a bayesian approximation: Representing model uncertainty in deep learning*, in *33rd International Conference on Machine Learning*, New York U.S.A. (2016), *Proceedings of Machine Learning Research. Vol. 48*, M.F. Balcan and K.Q. Weinberger eds., MLR Press, New York U.S.A. (2016), pg. 1050, http://proceedings.mlr.press/v48/gal16.html.

[109] A. Kendall and Y. Gal, *What uncertainties do we need in bayesian deep learning for computer vision?*, in *Advances in Neural Information Processing Systems. Vol. 30 (NIPS 2017)*, Long Beach U.S.A. (2017), I. Guyon et al. eds., Curran Associates, Inc., Red Hook U.S.A. (2017), pg. 5574, https://proceedings.neurips.cc/paper/2017/file/2650d6089a6d640c5e85b2b88265dc2b-Paper.pdf.

[110] L. Perreault Levasseur, Y.D. Hezaveh and R.H. Wechsler, *Uncertainties in Parameters Estimated with Neural Networks: Application to Strong Gravitational Lensing*, *Astrophys. J. Lett.* **850** (2017) L7 [arXiv:1708.08843] [INSPIRE].