Deducing Neutron Star Equation of State from Telescope Spectra with Machine-learning-derived Likelihoods

Delaney Farrell,[1] Pierre Baldi,[2] Jordan Ott,[2] Aishik Ghosh,[3,4] Andrew W. Steiner,[5,6]
Atharva Kavitkar,[7] Lee Lindblom,[8] Daniel Whiteson,[3] and Fridolin Weber[1,8]

[1]*Department of Physics, San Diego State University, San Diego, CA 92115, United States*
[2]*Department of Computer Science, University of California Irvine, Irvine, California 92697, USA*
[3]*Department of Physics and Astronomy, University of California Irvine, Irvine, California 92697, USA*
[4]*Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*
[5]*Department of Physics and Astronomy, University of Tennessee, Knoxville, Tennessee 37996, USA*
[6]*Physics Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, USA*
[7]*Department of Computer Science, TU Kaiserslautern, Germany*
[8]*Center for Astrophysics and Space Sciences, University of California at San Diego, San Diego, CA 92093, United States*

The interiors of neutron stars reach densities and temperatures beyond the limits of terrestrial experiments, providing vital laboratories for probing nuclear physics. While the star's interior is not directly observable, its pressure and density determine the star's macroscopic structure which affects the spectra observed in telescopes. The relationship between the observations and the internal state is complex and partially intractable, presenting difficulties for inference. Previous work has focused on the regression from stellar spectra of parameters describing the internal state. We demonstrate a calculation of the full likelihood of the internal state parameters given observations, accomplished by replacing intractable elements with machine learning models trained on samples of simulated stars. Our machine-learning-derived likelihood allows us to perform *maximum a posteriori* estimation of the parameters of interest, as well as full scans. We demonstrate the technique by inferring stellar mass and radius from an individual stellar spectrum, as well as equation of state parameters from a set of spectra. Our results are more precise than pure regression models, reducing the width of the parameter residuals by 11.8% in the most realistic scenario. The neural networks will be released as a tool for fast simulation of neutron star properties and observed spectra.

Contents

I. INTRODUCTION

Neutron stars are valuable astrophysical laboratories for studying matter under extreme conditions. With masses generally between 1 to 2 $M_\odot$ and radii between 10 and 15 km, the inner regions of these neutron-rich stars can reach density regimes well beyond those accessible in terrestrial laboratories. Matter at such high densities can potentially experience transitions to stable but unusual states of matter such as exotic baryons made of hyperons and Δ isobars [1–5]; deconfined up, down, and strange quarks [6, 7]; color superconducting phases [8–10]; or Bose-Einstein condensates made of negatively charged pions or K^- mesons [11–15]. A better understanding of the internal composition of these stars would shed light on many areas of current interest, including various astrophysical phenomena such as core-collapse supernovae [16] and binary star mergers [17], nuclear laboratory physics, QCD and relativistic gravity, as well as the early Universe. A long-standing issue in experimental and theoretical astrophysics is the determination of the equation of state (EOS) of matter within the cores of neutron stars, which describes the underlying relationship between pressure P and energy density ϵ. Theoretical EOS models make various assumptions regarding the state of matter within neutron stars' interiors, resulting in widely varying relationships between pressure and density.

The interiors of neutron stars are not available for direct observation. However, observational data from these stars, such as electromagnetic emission, is growing rapidly from telescopes like the *Chandra* X-ray Observatory, the Neutron star Interior Composition Explorer (NICER), and the Five-hundred-meter Aperture Spherical radio Telescope (FAST). The star's emitted spectrum is influenced by macroscopic stellar properties such as mass and radius, which are determined by the star's underlying EOS. The mass and radius, paired with other stellar parameters, control the emitted radiation, which when convoluted with the telescope response, determines the observed spectra. Therefore, in principle, the EOS can be inferred from observations of stellar spectra. In practice, crucial elements of the likelihood are not tractable, preventing straightforward inference. For example, while the mass-radius relation can be calculated from the EOS using the relativistic stellar structure equations (SSEs), the equations cannot be trivially inverted[18–24]. The task is further complicated by the small number of neutron star observations and the relatively large uncertainty of the individual measurements. Extraction of the maximal information content, with robust uncertainties, is therefore of paramount importance.

Recently, machine learning tools have demonstrated breakthroughs in data analysis in the natural sciences [25], increasing the power of valuable data [26] while naturally propagating uncertainties [27]. Applications in particle physics have used larger and deeper neural networks powered by advances in hardware processing to tackle more complex and higher-dimensional tasks [28, 29]. In astrophysical contexts such as regression of neutron star parameters where the likelihood is not tractable and analytical inversion is not feasible, simulation-based inference techniques [30] have been explored. For example, recent work from Fujimoto *et al* [31, 32] demonstrated regression of neutron star matter EOS from mass-radius pairs, where networks are trained on samples of simulated stars and uncertainties are drawn from an ad-hoc Gaussian model. Morawski [33] also performed regression from mass-radius pairs to EOS assuming Gaussian uncertainty and attempted to reduce EOS parameterization dependence. Soma *et al* [34] regressed both EOS and mass-radius information through a set of connected modes: density values are used to determine pressure values using a deep neural network, and the regressed pressure values are then used to generate mass-radius pairs using a generative deep learning model. Ferreira [35] instead regressed EOS from radius and tidal deformation information using support vector machines (SVMs) and deep neural networks (DNNs). Our recent work [36] demonstrated regression of EOS directly from simulated X-ray spectra without collapsing to intermediate states of observable properties like mass and radius, and where uncertainty due to unknown values of other stellar parameters is fully propagated. This was the first demonstration that neural networks have the capacity to directly analyze high-dimensional astrophysical data such as X-ray spectra.

Machine learning is, however, capable of more than regression. In this paper, we introduce a novel approach to the inference of EOS parameters from neutron star X-ray spectra, in which the likelihood is made tractable by replacing the intractable elements with neural networks trained on samples of simulated stars. Rather than directly learning the entire likelihood [37] in one step, we leverage our knowledge of the problem by replacing only the crucial missing pieces, which focuses the learning task and allows the interpretation of network outputs as physically meaningful quantities. The resulting *machine-learning-derived likelihood* of observing a set of stellar spectra given EOS parameters allows for estimation of the EOS parameters via Bayesian *maximum a posteriori* and use of standard error estimation techniques. The derivation of this machine-learning-derived likelihood is shown schematically in Figure 1.

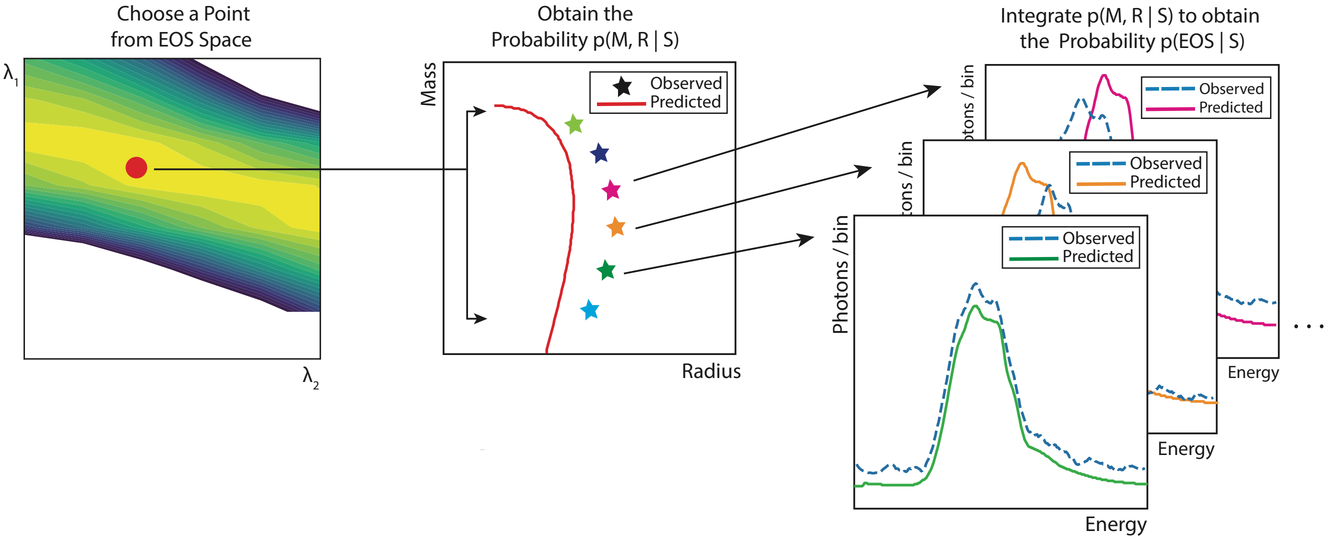

FIG. 1: Schematic diagram depicting the evaluation of the likelihood of producing a set of observed stellar spectra by comparing it to the predicted spectra along the mass-radius curve determined by EOS parameters λ_1, λ_2. Each value of the EOS parameters determines a curve in the mass-radius plane. Integrating along the curve, the probability of observing each star is evaluated as in Fig. 2.

This paper is outlined as follows. In Section II, we describe the physics of neutron star interiors. Section III gives an introduction to the relevant machine learning techniques. We describe the generation of our samples of simulated stars in Section IV, and the strategy for machine-learning-derived likelihoods in Section V. Sections VI and A demonstrate the likelihood calculation and extraction of parameter estimates for mass-radius information and EOS parameters, respectively. Sections B and C present a discussion of the results and future directions, respectively.

II. BACKGROUND

Observation of neutron stars has long served as a way to place constraints on our knowledge of superdense matter due to the theoretical connection between features deduced from observation (such as gravitational mass and radius) and the underlying EOS. The theoretical connection comes from the relativistic stellar structure equations, derived from Einstein's field equations, where the star's internal structure is used to derive the stellar properties of a compact object.

For spherically symmetric, non-rotating stars comprised of isotropic material in hydrostatic equilibrium, the stellar structure is compactly summarized in the Tolman-Oppenheimer-Volkoff (TOV) equation (in geometerized units $G = c = 1$):

$$\frac{dP}{dr} = -\frac{(\epsilon + P)(m + 4\pi r^3 P)}{r^2\left(1 - \frac{2m}{r}\right)}, \tag{1}$$

where m is the gravitational mass enclosed within a sphere of radius r, P is the pressure, and ϵ is the energy density. This equation creates a one-to-one map from the EOS to a mass-radius relation [23], where the inverse mapping would provide constraints on the EOS from mass and radius values. While inverting the relativistic SSEs has been demonstrated to be numerically intractable, mass and radius values determined by observation provide valuable constraints on EOS; observations of high-mass pulsars (such as PSR J0952-0607, where $M = 2.35 \pm 0.17\, M_\odot$ [38]) have ruled out previously popular EOS models which failed to predict masses above $2M_\odot$.

Observation of neutron star emission has long served as a way to constrain mass and radius for neutron stars (eg. [39, 40]), which in turn places constraints on the nuclear EOS. Some of the most precise constraints thus far come from observing thermal emission from quiescent low-mass X-ray binaries (qLMXBs). These binary systems are identified in well-studied globular clusters where distances, ages, and reddening are well-constrained [41]. Additionally, their low magnetic fields result in minimal effects on the radiation transport or temperature distribution on the star's surface [42–44], making them particularly desirable for placing strong constraints on neutron star structure.

Thermal emission from qLMXBs is observed with powerful telescopes such as NASA's *Chandra* X-ray Observatory, where high-resolution imaging and spectroscopy have provided powerful insight into neutron star properties such as cooling [45, 46], mass and radius limits [42], and binary mergers of exotic stars [47]. *Chandra*'s telescope contains a system of four pairs of mirrors that focus incoming X-ray photons to the Advanced CCD Imaging Spectrometer (ACIS), which measures the energy of each incoming X-ray. The observed spectrum then determines neutron star bulk properties like mass and radius through spectral fitting, where both the spectrum and corresponding instrument response are fit to an appropriate atmosphere model, where the surface composition is known or can be determined by the X-ray spectrum. An extensive list of such models exists in XSPEC [48], an X-ray spectral fitting package distributed and maintained by the aegis of the GSFC High Energy Astrophysics Science Archival Research Center (HEASARC). XSPECis a state-of-the-art tool to analyze X-ray data from *Chandra* and other spectrometers like *NICER*, *Nustar*, and *XMM-Newton* - making it a valuable resource for inference of neutron star properties.

III. MACHINE LEARNING

Machine learning is a branch of computer science, and the main component of today's artificial intelligence, where computers learn to compute useful functions from data. In deep learning, a modern re-branding of neural networks and today's major branch of machine learning, networks of simple neurons connected by adjustable synaptic weights are used to implement flexible classes of functions. In a typical case, the parameters of such models, i.e. the synaptic weights, are adjusted incrementally from the training data via a stochastic gradient descent process in order to minimize some error function. For instance, in the case of a regression problem, neural networks can be trained to minimize the standard least-squared error over the training data. This can be viewed as a generalization of standard linear regression, where the linear model is replaced by a more complex, non-linear, neural network model. The least-squares error can be viewed as the negative log-likelihood of the data under a Gaussian noise model. The neural network learns to predict the parameters of the negative log-likelihood function, i.e. the mean of the Gaussian distribution in the standard regression framework.

Machine learning, and in particular deep learning, have many applications in science [25] and have greatly benefited from the increase in computing power over the last few decades, especially in the form of GPUs (Graphical Processing Units) which are well suited for the matrix-vector multiplications - the computational workhorse of neural networks. Deep learning methods can flexibly handle data sets, models, and dimensions that range from very small to very large. For instance, deep learning models with millions, or even billions, of adjustable parameters, are not uncommon. These methods can handle and leverage raw data, without the need for pre-processing using heuristic or manual simplifications, which often discard valuable information. In physics applications, this often leads to significant performance improvements in terms of accuracy, detection rates, and so forth (e.g. [26, 49, 50]). In the rest of this work, we use deep learning methods for regression to deterministically compute the parameters of complex functions needed to calculate likelihoods.

IV. TRAINING DATA

Samples of simulated stars are used both to train and evaluate network performance. The samples used in these studies were originally produced for Ref. [36]; we use the identical samples here to allow direct comparison to previous work. Each simulated star is described by two parameters of interest and three additional physical stellar properties. The parameters of interest are mass M and radius R, derived from EOS variations using the stellar structure equations. The other three quantities, deemed nuisance parameters, affect the stellar spectrum but are not related to the EOS: the effective temperature of the star's surface, T_{eff}, the distance to the star, d, and the hydrogen column, N_H which parameterizes the reddening of the spectrum by the interstellar medium. All five parameters determine the simulated X-ray spectra, generated using XSPEC. Details of each data generation step are outlined below.

A. Generation of Equation of State

As described in Ref. [36], the EOS models used to generate many simulated neutron stars are variations of the relativistic, non-linear mean-field model GM1L [51]. The parameterization of GM1L used in these studies accounts only for protons and neutrons; the corresponding saturation properties of this parametrization are shown in Table 1 of Ref. [36], which follow current constraints from nuclear experiments and astrophysical observations [52]. In order to represent the EOS in a compact fashion and limit parameters of interest, we represent GM1L in a parametric representation based on spectral fits; the process for constructing such fits is outlined in detail in Refs. [53] and [21].

TABLE I: Description of "true", "tight", and "loose" nuisance parameter (NP) scenarios. Shown are the width of each Gaussian distribution representing the prior knowledge of each NP. For distance and N_H, width is relative; for $\log(T_{\text{eff}})$, it is absolute. See text for details and references.

Nuis. Param.	True	Tight	Loose
Distance	exact	5%	20%
Hydrogen Column N_H	exact	30%	50%
$\log(T_{\text{eff}})$	exact	± 0.1	± 0.2

As in Ref. [36], the parametric representation of GM1L is given by the coefficients of order two spectral expansion: λ_1 and λ_2. We randomly vary the two original coefficients to create 10^4 unique EOS variations in order to generate the many models needed for training the networks discussed below. Each EOS variation was then used to generate a mass-radius relation using the TOV equation (1). From each mass-radius relation, 100 (M, R) pairs whose mass falls in the physical range of neutron star masses (1 to 3 $M_\odot$) are selected for training and validation.

B. Modeling Spectra

Generation of a simulated neutron star X-ray spectrum requires the stellar mass and radius, generated from a particular EOS model as described above, as well as the three nuisance parameters. Beyond spectral fitting, the XSPEC program [48] can create simulated spectra via the `fakeit` command. The command requires a specified theoretical model and telescope response matrix. As described in greater detail in Ref. [36], the theoretical hydrogen atmosphere model `NSATMOS` [54] was used, paired with the *Chandra* telescope response specified in Ref. [54] which describes the instrument response and telescope effective area.

C. Nuisance Parameters

The `NSATMOS` model implemented in XSPEC requires five inputs for each simulated spectrum. The first two, gravitational mass M in units of $M\odot$ and radius R in units of kilometers, are taken from mass-radius relations from the GM1L EOS variations. The remaining three parameters are the effective temperature of the star's surface T_{eff}, the distance to the star, d, and the hydrogen column density, N_H. These three additional parameters are referred to as nuisance parameters (NPs) in Ref. [36] and are given values drawn from reasonable physical limits.

To determine the physical limits of the distance to the star and the hydrogen column density N_H, we turn to Table 1 in Ref. [40]. The distances typically range between 2 and 10 kpc, and hydrogen columns lie between 0.2 and 5×10^{21} cm^{-2}. From Table 3 in Ref. [55], effective temperatures at the surface typically lie between 50 and 200 eV, or from 6×10^5 and 2.4×10^6 K, with temperatures in the core increasing by a few orders of magnitude.

Examples of generated X-ray spectra are shown in Ref. [36]. The three NPs, which we will refer collectively to as ν, directly impact the simulated spectra and the uncertainty in the spectral fitting process. In order to propagate the uncertainty from these parameters and gauge our networks' sensitivity to each parameter, we define three example scenarios of uncertainties: "true", "tight", and "loose". The three scenarios describe the quality of prior information on the NP values for each star.

In the "true" scenario, the NPs are set to the true value chosen from the physical range used to generate the spectra, such that the NP prior is essentially a delta function. In the "tight" scenario, the uncertainty is described as a narrow Gaussian for each NP, with distance having a width of 5%, hydrogen column having a width of 30%, and $\log(T_{\text{eff}})$ having a width of 0.1. In the "loose" scenario, the uncertainties are described by a wider Gaussian, with distance having a width of 20%, hydrogen column having a width of 50%, and $\log(T_{\text{eff}})$ having a width of 0.2. Table I shows the width of the prior information of the three NPs for the three uncertainty scenarios.

V. MACHINE-LEARNING DERIVED LIKELIHOOD CALCULATION

If the likelihood $p(x|\theta)$ of observing some data x for given values of the theoretical model parameters θ were known, estimating the value of θ for a given x would be a straightforward task. In a Bayesian approach, one would use *maximum a posteriori* (MAP) to find $\hat{\theta}$, the value of θ which maximizes the product of the likelihood and the prior $p(\theta)$.

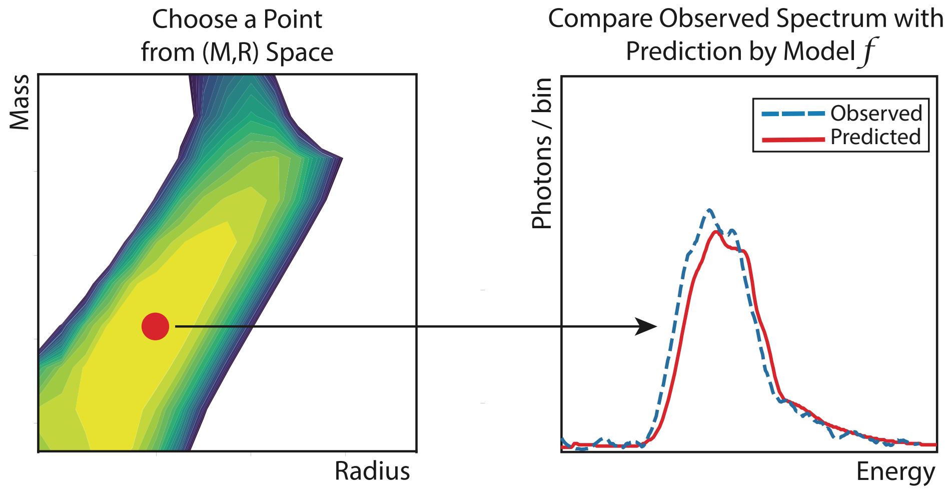

FIG. 2: Schematic diagram depicting the evaluation of the likelihood of producing on the observed stellar spectrum by comparing it to the predicted spectrum for assumed values of stellar mass and radius. Varying the assumed mass and radius allows for an evaluation of the likelihood across the (M, R) parameter space.

In the case of neutron star EOS estimation, the likelihood $p(S|\lambda_1, \lambda_2)$ of observing a set of stellar spectra S for a given choice of EOS parameters λ_1, λ_2 is not tractable. Instead, simulation-based inference [30] techniques often circumvent the need for a closed-form likelihood by approximating $p(S|\lambda_1, \lambda_2)$ using density estimation with simulated samples. But accurately estimating the density in high-dimensional spaces such as those for stellar spectra, with dimensionality of $\approx \mathcal{O}(1000)$, would require prohibitively large samples of simulated events. Instead, we use machine learning to estimate the likelihood. But rather than estimating the entire likelihood in one step, our machine-learning-derived likelihood calculation focuses the statistical power where it is most needed for a given set of observations by combining the available analytic components with neural networks to replace the intractable components.

For the estimation of parameters such as EOS or mass and radius, many elements of the likelihood calculation are known, such as the Poisson fluctuations within each energy bin of a stellar spectrum. However, there are two elements without closed-form expressions that require machine-learning assistance. We overview those intractable components here briefly and provide more detail below.

The first missing piece is a prediction for the expected photon rates observed in a telescope given stellar parameters M and R and nuisance parameters ν. Below, we train a neural network to estimate this quantity and demonstrate its application by doing MAP estimates of M and R, in which we integrate over the nuisance parameters ν.

The second missing piece is a prediction for the mass-radius values allowed by given EOS parameters λ_1, λ_2. We train a neural network to output the allowed radius for a given stellar mass when provided with the EOS parameters. This allows integration over the $M - R$ curve defined by the EOS.

Together, these two machine-learned elements allow for a tractable calculation of the likelihood $p(S|\lambda_1, \lambda_2)$ and provide a MAP estimate of λ_1, λ_2 for a given set of spectra S. We describe each piece in turn below.

VI. STELLAR MASS AND RADIUS INFERENCE

The mass and radius of a neutron star can be estimated from its spectrum s if one can calculate and maximize $p(s|M, R)p(M, R)$ for a fixed s. This requires access to $p(s|M, R)$, which depends on $p(s|M, R, \nu)$, the likelihood to produce this spectrum s given the full set of stellar parameters including the nuisance parameters ν:

$$p(s|M, R) = \int d\nu \; p(s|M, R, \nu) \; p(\nu),$$

If the spectra consists of a set of energy bins of telescope photon counts,

$$s = (N_1^\gamma, N_2^\gamma, ..., N_{n_{\text{bins}}}^\gamma),$$

then the joint likelihood $p(s|M,R,\nu)$ over the bins can be written as

$$p(s|M,R,\nu) = \prod_{j}^{n_{\rm bins}} {\rm Pois}(N^\gamma_j, \mu_j(M,R,\nu)),$$

where μ_j is the expected number $\langle N^\gamma_j \rangle$ of photons in bin j of n bins:

$$\mu(M,R,\nu) = (\langle N^\gamma_1 \rangle, \langle N^\gamma_2 \rangle, ..., \langle N^\gamma_{n_{\rm bins}} \rangle),$$

determined by M, R, ν via complex physics of the stellar emission model as well as the telescope response. In principle, for this specific scenario, this function is contained within XSPEC, but is not exposed to the user in a convenient fashion which would allow for a rapid evaluation of many points, as needed for this application. More generally, one may want to learn a function in scenarios where no function is available, or where the training sample contains examples generated from a mixture of models where no single simple function can describe all samples.

When we do not have access to a simple expression for $\mu(M,R,\nu)$, it is possible to train a neural network to learn a function $f[M,R,\nu]$

$$f[M,R,\nu] \rightarrow (\frac{dN^\gamma_1}{dt}, \frac{dN^\gamma_2}{dt}, ..., \frac{dN^\gamma_{n_{\rm bins}}}{dt}),$$

such that $\mu(M,R,\nu)$ can be estimated from $f(M,R,\nu)$ scaled by the observation time Δt,

$$\mu \approx f[M,R,\nu]\Delta t.$$

The likelihood $p(s|M,R,\nu)$ can then be estimated as

$$p(s|M,R,\nu) = \prod_{j}^{n_{\rm bins}} {\rm Pois}(N^\gamma_j, \mu_j = f[M,R,\nu]_j \Delta t),$$

where the final likelihood can be obtained by marginalizing over the nuisance parameters:

$$p(s|M,R) = \int d\nu \; p(s|M,R,\nu) \; p(\nu) = \int d\nu \prod_{j}^{n_{\rm bins}} {\rm Pois}(N^\gamma_j, \mu_j = f[M,R,\nu]_j \Delta t) p(\nu). \qquad (2)$$

This estimate allows for a scan of the M, R plane for the values which maximize $p(s|M,R)p(M,R)$ to find the MAP estimate for (M,R) given a fixed s. Schematically, this process is shown in Fig 2.

A. Learning the Model f for Stellar Spectra

The estimated likelihood requires learning a function $f[M,R,\nu]$ which produces the expected stellar spectrum in each bin. This function is modeled by a deep neural network with five inputs (M, R, and the three components of ν: effective temperature, distance, and N_H) and 250 outputs, one for each of the spectral bins.

The network architecture includes two input branches, one to process the mass and radius, and another to process the nuisance parameters. Each branch contains a series of nine layers, with 2048 nodes and leaky ReLU activation, that process its inputs in isolation. Following these initial layers, the output from the branches is combined together, forming a single volume with all information. This grouping is then passed to another successive set of nine layers which produce the generated spectra. The choice of hyperparameters has a significant impact on performance of this network. A complete list of hyperparameters tried is given in

; the architecture used for model f was the configuration with the best performance on the validation set.

Generating the spectra is formulated as a supervised learning problem, where the network learns to minimize the error between the true spectra and its predictions for a set of simulated examples generated by XSPEC. The weights are updated with gradient descent via backpropagation. The Huber loss function is used, which is a standard loss

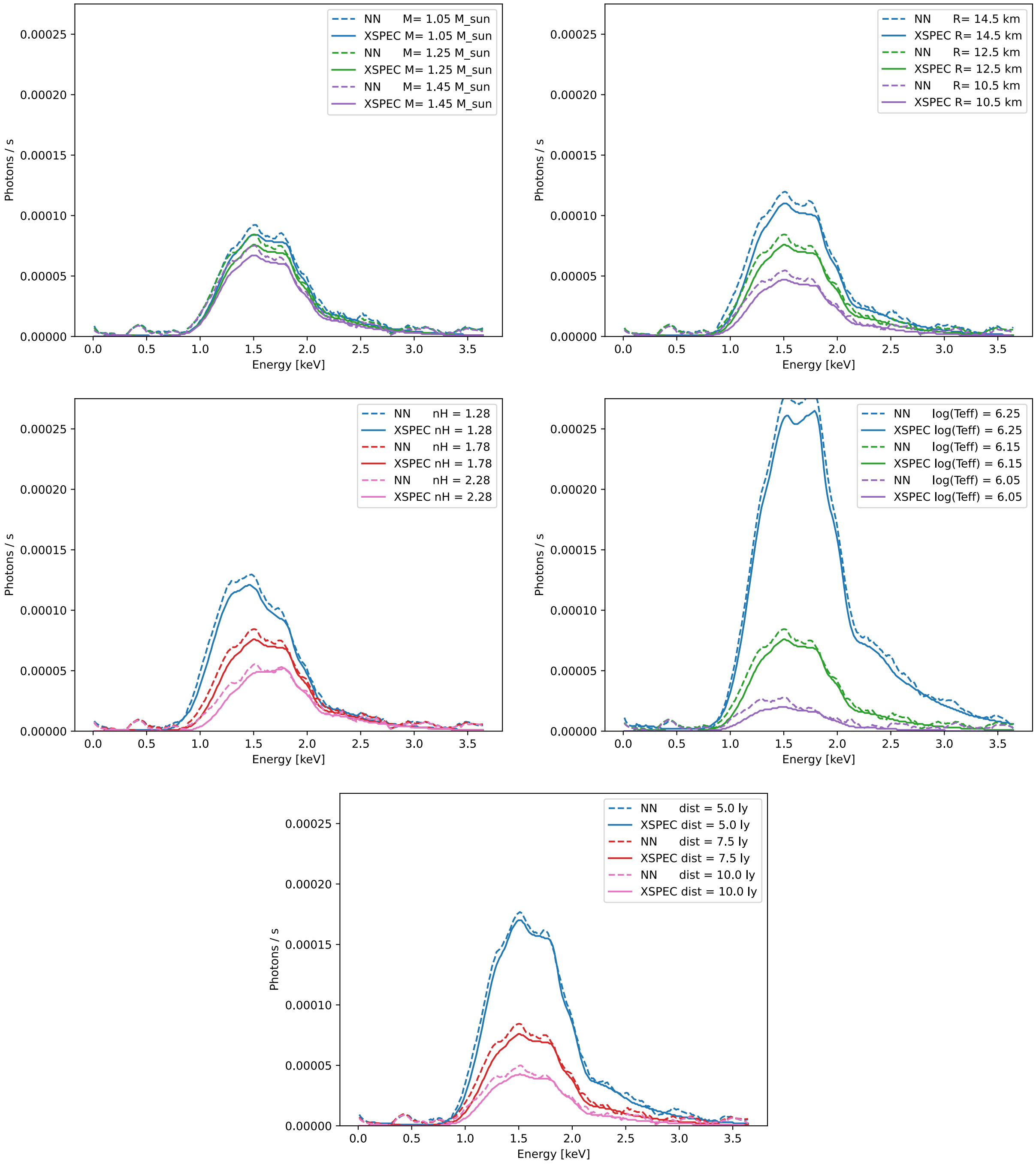

FIG. 3: Comparison of neutron star X-ray spectra predictions (dashed) from our network $f[M, R, \nu]$ described in the text, as compared to training data generated by XSPEC (solid). Each pane shows the expected rate of photons ($\frac{dN_1^\gamma}{dt}$) in Chandra per energy bin, for variations of the parameters of interest (mass M, radius R) as well as for variations of the nuisance parameters ν (n_H, $\log(T_{\text{eff}})$, distance).

function for regression robust to outliers, and the Adam optimizer computes gradients and schedules the backward passes.

Figure 3 shows several examples of generated spectra. Each subplot contains generated results using various values of mass, radius, and nuisance parameters. The predictions are shown with their corresponding version from XSPEC. The predictions track the XSPEC values remarkably well across a range of parameter values. In general, we notice a slight overprediction by the network; below, we estimate the potential bias due to this overprediction and find it to

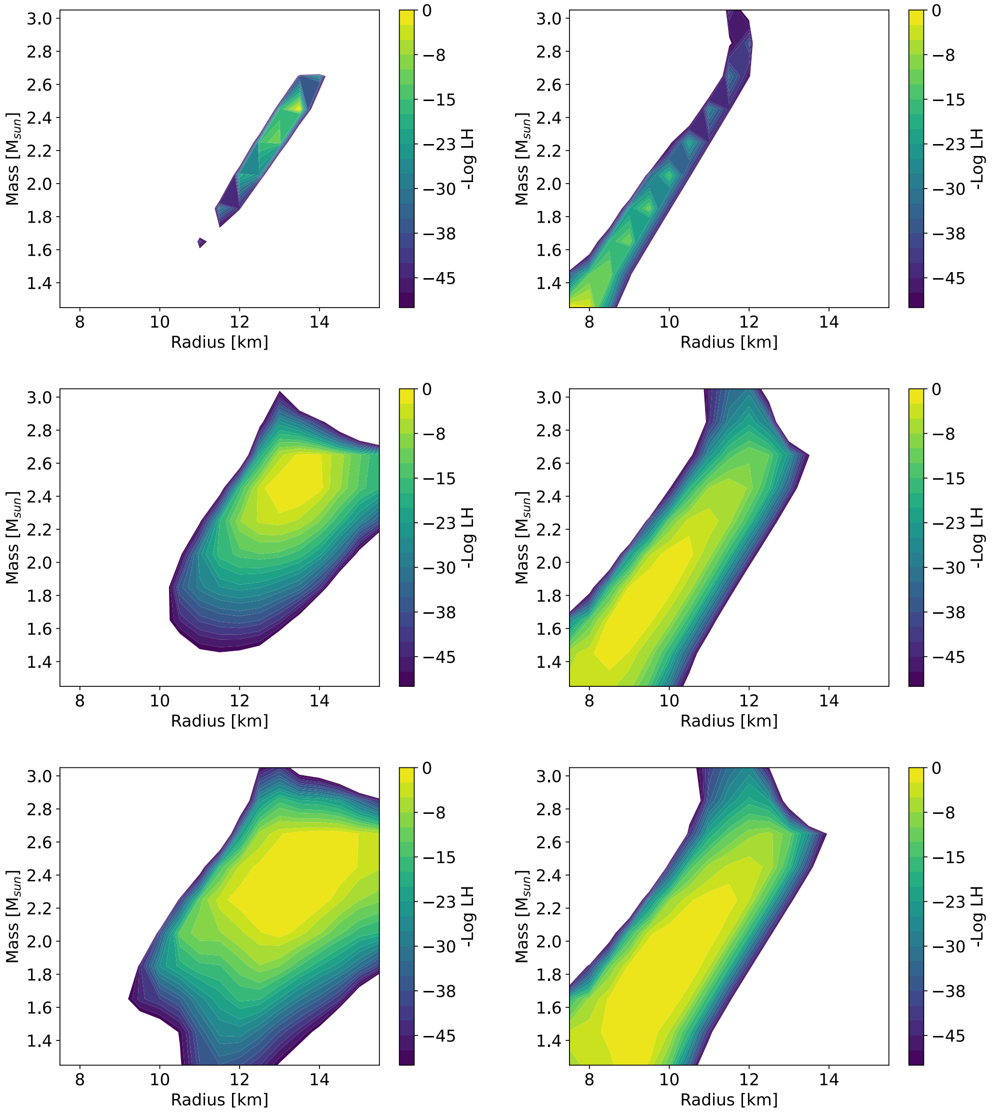

FIG. 4: Scans of the likelihood for two example stellar spectra s (left, right) versus stellar mass and radius. Top demonstrates the ideal nuisance parameter (NP) conditions where the NPs are fixed to their true values. For the same simulated observed spectra, center shows a more realistic "tight" scenario , and bottom shows a "loose" scenario in which the NPs are not well constrained by priors. In the "loose" and "tight" scenarios, dependence on the nuisance parameters has been integrated out as described in the text.

be negligible compared to statistical uncertainties and other systematic uncertainties.

1. Results

The machine-learning-derived likelihood is tractable, allowing for a scan of $p(s|M, R)$ for individual stars. Figure 4 shows examples of two individual simulated stars under the three nuisance parameter scenarios.

The likelihood can then be used to produce estimated values of neutron star mass and radius for a given spectrum, after marginalizing over the nuisance parameters. The mass-radius plane is scanned using adaptive optimization to find values that maximize the product of the likelihood and the prior.

We assess the performance of this method to produce likelihood estimates, ML-Likelihood$_{\mathrm{M,R}}$, by calculating the residual between the true values of the mass and radius and the estimates produced by our method. As benchmarks, we compare the residuals to those generated by XSPEC itself, which has access to the true likelihood used to generate the simulated samples, as well as a regression-based method, MR_Net, described in Ref. [36]. Figure 5 and Table II shows a comparison of the results.

It is striking that ML-Likelihood$_{\mathrm{M,R}}$ outperforms MR_Net dramatically, despite being trained on the same dataset. ML-Likelihood$_{\mathrm{M,R}}$ benefits from the physics knowledge encoded in the approximate likelihood, which requires ML solutions only for a specific sub-task, rather than having to blindly learn the entire problem, as MR_Net must. ML-Likelihood$_{\mathrm{M,R}}$ performs essentially as well as XSPEC, even having smaller residuals when nuisance parameters have the most uncertainty, despite not having access to the explicit likelihood used to generate the data, as XSPEC does. Note that the performance comparison here highlights a *practical* difference between our method and use of XSPEC, rather than a *principled* difference in the methods. That is, our method of replacing the difficult calculational step with a learned ML model allows for convenient and rapid evaluation of the likelihood over many values of the nuisance parameters, enabling us to marginalize over them, which explains the improved relative performance. While in principle one could perform the same operation with XSPEC, its lack of a convenient programmable interface makes this impractical. The practical distinction is crucial, however, as it allows for greater general flexibility, such as interpolation across several models, or in application to the broader EOS inference problem described below, where likelihoods are not just hidden behind inconvenient interfaces, but completely unavailable.

TABLE II: Performance of our estimation of neutron star mass (left) and radius (right) using an approximate likelihood which incorporates neural networks, ML-Likelihood$_{\mathrm{M,R}}$, in comparison to the performance of a pure regression network, MR_Net [36] and the XSPEC tool. Shown are the mean (μ) and standard deviation (σ) of the residual distributions under three scenarios of nuisance parameter uncertainties. In the “true” case, the NPs are fixed to their true values; in the “tight” and “loose” cases, they are drawn from narrow or wide priors, respectively; see text for details.

	Nuis.	Mass		Radius	
Method	Params	μ	σ	μ	σ
ML-Likelihood$_{\mathrm{M,R}}$	True	0.11	0.47	0.28	1.43
XSPEC	True	-0.01	0.50	0.23	1.44
MR_Net	True	-0.14	0.93	-0.7	2.80
ML-Likelihood$_{\mathrm{M,R}}$	Tight	0.15	0.50	0.57	1.87
XSPEC	Tight	0.06	0.73	0.24	2.61
MR_Net	Tight	0.17	1.06	0.06	3.52
ML-Likelihood$_{\mathrm{M,R}}$	Loose	0.15	0.58	0.91	2.42
XSPEC	Loose	0.18	0.86	-0.15	4.32
MR_Net	Loose	0.28	1.29	0.14	4.93

Appendix A: Equation of State Inference

The ultimate goal is to estimate the EOS parameters (λ_1, λ_2) given a set of spectra $S = (s_1, s_2..., s_{n_{\mathrm{stars}}})$. In principle, this would be straightforward if one could evaluate $p(S|\lambda_1, \lambda_2)p(\lambda_1, \lambda_2)$, which would allow for maximization to find an estimate for λ_1, λ_2 for a fixed S.

We begin with the assumption that the EOS parameters have a uniform prior within their physical boundaries of $\lambda_1 \in [4.75, 5.25], \lambda_2 \in -[1.85, -2.05]$ (see Fig 3 of Ref. [36]). The remaining step is evaluating $p(S|\lambda_1, \lambda_2)$.

First we express the probability over the entire set S as the joint probability for each star s_i:

$$p(S|\lambda_1, \lambda_2) = \prod_{i}^{n_{\mathrm{stars}}} p(s_i|\lambda_1, \lambda_2).$$

The obstacle is that we do not know how to evaluate $p(s|\lambda_1, \lambda_2)$, only $p(s|M, R)$, which depends on stellar parameters M and R. Linking these expressions is not trivial, as the EOS parameters λ_1, λ_2 do not uniquely determine stellar parameters M and R, instead they only determine the M-R *relation.* That is, each point in (λ_1, λ_2) space specifies a curve in M-R space. The solution is to integrate over the M-R curve allowed by the EOS parameters λ_1, λ_2. This is most directly accomplished by expressing the integral over the mass-radius plane, constrained by a delta function which traces out the M-R curve determined by the EOS parameters λ_1, λ_2:

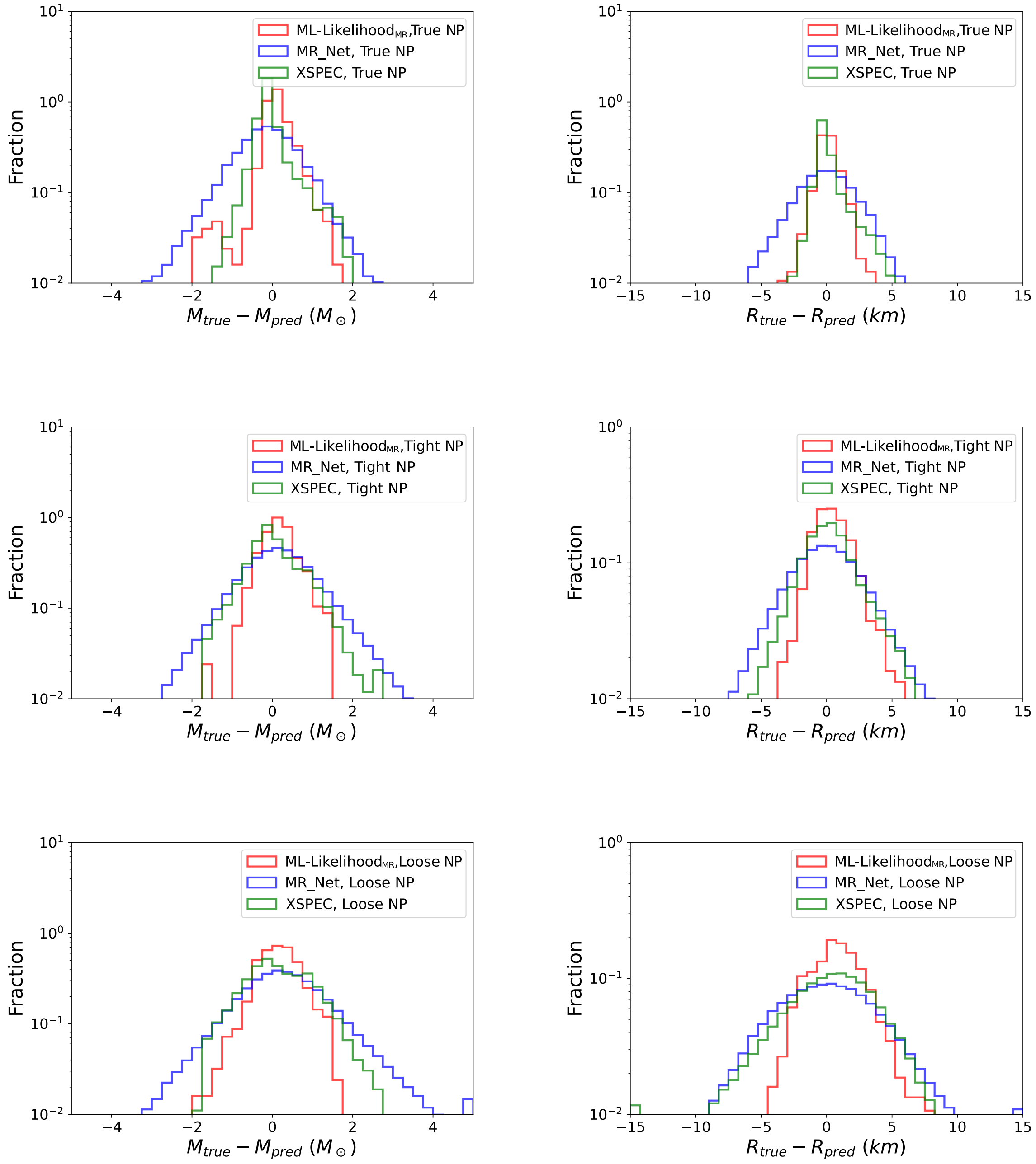

FIG. 5: Performance of our estimation of neutron star mass (left) and radius (right) using an approximate likelihood which incorporates neural networks, ML-Likelihood$_{\rm M,R}$, in comparison to the performance of a pure regression network, MR_Net [36] and the XSPEC tool. Shown is the residual, the difference between the true and predicted values, for three scenarios of nuisance parameter uncertainties. In the "true" case, the NPs are fixed to their true values; in the "tight" and "loose" cases, they are drawn from narrow or wide priors, respectively; see text for details.

$$p(s|\lambda_1, \lambda_2) = \int dM dR \; p(M, R|\lambda_1, \lambda_2) p(s|M, R),$$

where $p(M, R|\lambda_1, \lambda_2)$ describes the allowed M-R relation given the EOS specified by λ_1, λ_2, as

$$p(M,R|\lambda_1,\lambda_2) = \delta(h_\lambda[M]-R)\ \ p(M|\lambda_1,\lambda_2),$$

where $h_\lambda[M] \to R$ is a function that gives the allowed value of R for a value of M, determined by the EOS parameters λ_1, λ_2. The function $h_\lambda[M]$ encodes all of the physics which translates the EOS into stellar mass and radius, and is not available analytically or tractable numerically. It is possible, however, to train a neural network to learn this function, as we do below. Assuming $h_\lambda[M]$ is available, we choose to integrate over mass, as each mass is mapped to a unique R; the same is not true for scanning in R, see Figure 6.

The delta function reduces the double integral in M and R to a single integral over mass:

$$p(s|\lambda_1,\lambda_2) = \int dM\ p(M|\lambda_1,\lambda_2)\ p(s|M,R=h_\lambda[M]),$$

where the range of the mass integral is limited to the physical region, from $1.2M_\odot$ to $1.6-3.25M_\odot$, depending on the radius, see Figure 6. This allows us to write an expression for the joint probability over the set of stars:

$$p(S|\lambda_1,\lambda_2) = \prod_i^{n_{\rm stars}} \int dM_i\ p(M_i|\lambda_1,\lambda_2)\ p(s_i|M_i,R_i=h_\lambda[M_i]).$$

We have now expressed the likelihood $p(S|\lambda_1,\lambda_2)$ in terms of the likelihood $p(s|M,R)$, which we previously learned to calculate. The equation for $p(s|M,R)$ now allows for the expression of a joint likelihood over the stars and the bins:

$$L_S(\lambda_1,\lambda_2) = p(S|\lambda_1,\lambda_2) = \prod_i^{n_{\rm stars}} \int dM_i\ p(M|\lambda_1,\lambda_2) \int d\nu \prod_j^{n_{\rm bins}} {\rm Pois}(N^\gamma_{ij}, \mu_{ij}(M_i,R_i=h_\lambda[M_i],\nu) p(\nu),$$

where we can replace each of the μ_{ij} as we did above with $f[M,R,\nu]\Delta t$

$$L_S(\lambda_1,\lambda_2) = p(S|\lambda_1,\lambda_2) = \prod_i^{n_{\rm stars}} \int dM_i\ p(M|\lambda_1,\lambda_2) \int d\nu \prod_j^{n_{\rm bins}} {\rm Pois}(N^\gamma_{ij}, \mu_{ij} = f[M_i,h_\lambda[M_i],\nu]_j \Delta t) p(\nu). \tag{A1}$$

This expression can be evaluated, assuming one can learn a function $h_\lambda[M] \to R$. The determination of the likelihood $L_S(\lambda_1,\lambda_2)$ is shown schematically in Fig. 1.

1. Learning the Model $h_\lambda[M]$ for Stellar Radius

The approximate likelihood above requires learning a function $h_\lambda[M]$ which estimates the stellar radius for a given stellar mass as determined by the EOS parameters λ_1, λ_2. Note that one could equivalently estimate the mass from the radius, but this has the additional complication of degenerate outputs for some radii, see Fig. 6. It is important here to note that model $h_\lambda[M]$ is trained on M-R relations created by equations of state that do not feature a phase transition; parameterizing the M-R relation as a function of $R = h_\lambda[M]$ may miss details stemming from more exotic features, like those caused by a strong, first-order phase transition, in the EOS.

We model $h_\lambda[M]$ with a deep neural network comprising 10 hidden layers with 32 nodes each and ReLU activation. The output layer is a single node with linear activation, which is standard for regression. The number of hidden layers, their widths, and activation functions were optimized for the functionality of $h_\lambda[M]$; the relatively small width of 32 nodes and ReLU activation were found to perform well. The network was trained with Mean Squared Error (MSE) loss and the Adam optimizer. Figure 6 demonstrates how the network $h_\lambda[M_i] \to R$ performs for a few example values of the EOS parameters λ. Generation of the training data as described above required approximately 24 hours of CPU time for 10^6 stars; in comparison, evaluation of the network $h_\lambda[M]$ generates 10^6 stellar radii in 400 ms, a relative speed enhancement of 10^5.

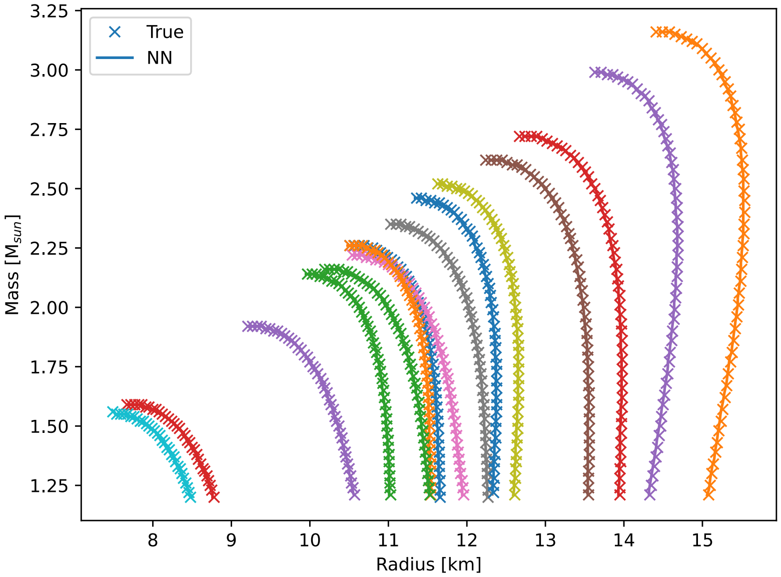

FIG. 6: Relationship between neutron star mass and radius, as determined by equation of state parameters λ_1, λ_2. Each color represents a single choice of EOS parameters, which determine a curve in the mass-radius plane. Individual calculations as described in the text are shown (crosses), as compared with the output of a neural network function $h_\lambda[M]$ (solid line), which estimates the radius corresponding to an input value of M as determined by the EOS parameters.

2. Results

The two networks that model the missing functions f and h allow for an approximate evaluation of the likelihood, Eq. A1 as a function of the EOS parameters. Figure 7 shows examples of two individual sets of simulated stars under the three nuisance parameter scenarios.

To estimate the EOS parameters from a fixed set of stellar spectra, the likelihood is maximized via a course scan over EOS parameter space followed by the use of an optimization algorithm for a more refined location of the optimal EOS parameters. Each evaluation of the likelihood involves nested loops over the stars, an integral over possible masses, and a loop over the spectral bins. Performance of ML-Likelihood$_{\mathrm{EOS}}$ and comparison with benchmarks are shown in Fig. 8 and Table III. We note that the data used in evaluations are generated via XSPEC, not from the models f and h, allowing for a test of the fidelity of the machine-learned models. Experiments in which simulated spectra are generated using the models f and h show equivalent performance, indicating that any bias due to mis-estimation by f or h is negligible in this context.

Appendix B: Discussion

The results above demonstrate that machine-learning-derived likelihoods are useful statistical tools, allowing for traditional inference such as parameter estimation for quantities of interest (eg stellar M and R) as well as profiling over nuisance parameters (eg stellar distances and temperatures).

In the case of M,R-estimation for an individual star, the performance of the ML-Likelihood$_{\mathrm{M,R}}$ method matches the performance of XSPEC when the nuisance parameters are known. This is an important validation of the technique, as the simulated samples are generated by XSPEC and so its internal likelihood estimation represents something of an upper bound on possible performance. Though XSPEC can provide point estimates and other analysis, ML-Likelihood$_{\mathrm{M,R}}$ in this case is valuable as a building block for further analysis, as XSPEC does not provide an efficient interface to its internal calculations. For example, in the cases where nuisance parameters weaken the inference, ML-Likelihood$_{\mathrm{M,R}}$ is able to improve on XSPEC's performance by marginalizing over the stellar nuisance parameters. Given access to the full likelihood, one could also choose to profile over the nuisance parameters. In addition, while XSPEC's inference is linked to a particular theoretical model, ML-Likelihood$_{\mathrm{M,R}}$ can be trained on a variety or mixture of models, providing a smooth interpolation between otherwise distinct conceptual approaches [56].

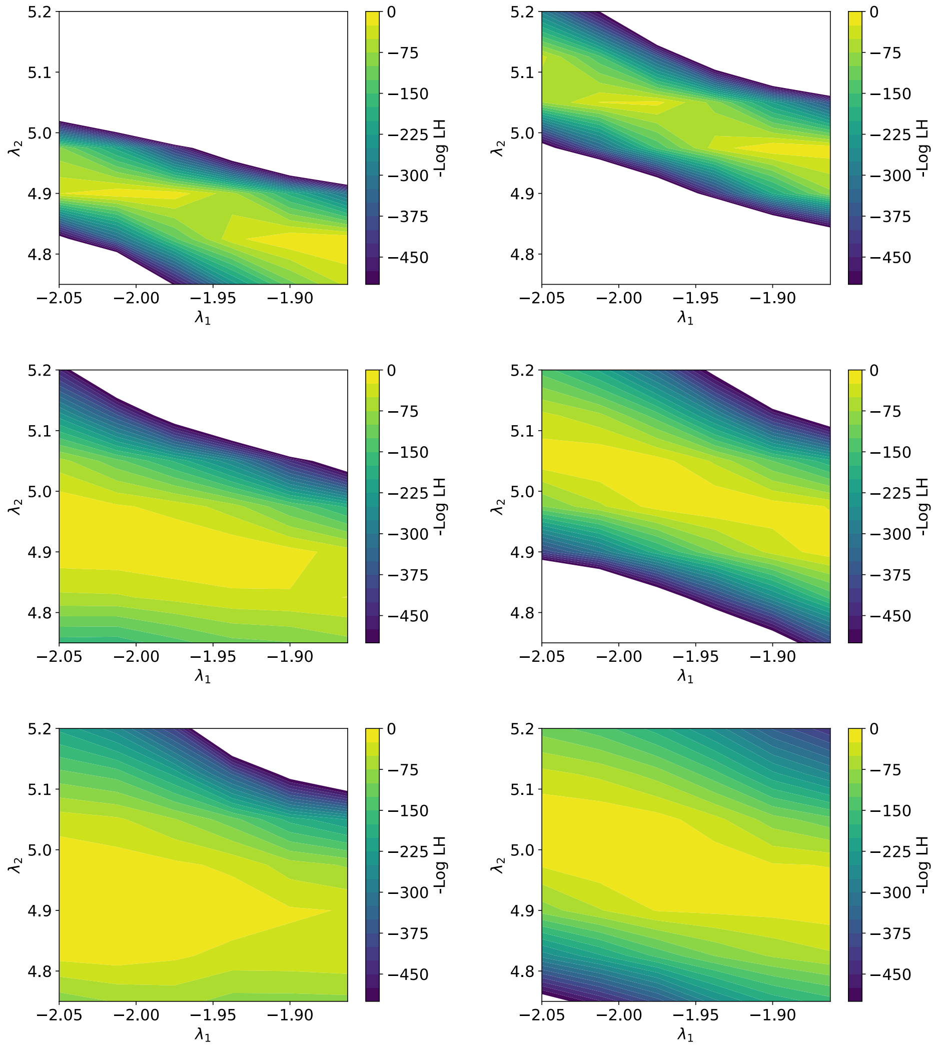

FIG. 7: Scans of the likelihood for two example sets of stellar spectra s (left, right) versus EOS parameters λ_1 and λ_2. Top demonstrates the ideal nuisance parameter (NP) conditions where the NPs are fixed to their true values. For the same simulated observed spectra, the center shows a more realistic "tight" scenario, and the bottom shows a "loose" scenario in which the NPs are not well constrained by priors. In the "loose" and "tight" scenarios, dependence on the nuisance parameters has been integrated out as described in the text.

The M, R-likelihood estimation is a building block toward the estimation of EOS parameters for sets of stars. In this case, as well, the likelihood provides for reliable inference of the EOS parameters, as demonstrated by the performance of ML-Likelihood$_{\text{EOS}}$. The residuals in this case again are narrower than the pure regression approach, nearly matching the performance of XSPEC in the true case, and exceeding its unmarginalized estimates in the realistic case where nuisance parameter uncertainty is important.

Our method uses machine learning to enable what is typically termed a *forward* process, in that it aids the calculation of the likelihood of experimental data from the parameters, rather than *backward* inference of the parameters from the data. In this sense, it can be considered a fast and flexible simulation tool. The neural networks developed for this work effectively enable end-to-end, fast, and convenient simulation of neutron star spectra for a range of EOS parameters and nuisance parameters, including the intermediate step of generating plausible neutron star properties (mass and radius) for a given set of EOS parameters. In the case where the true likelihood exists but is not made conveniently accessible, our approach provides a powerful and flexible new interface, even without speed enhancements. In the more general case, such as for EOS inference, the approach additionally allows for more rapid generation of simulated stellar masses and radii for specific EOS parameters without solving the complex sets of equations underlying the physical model.

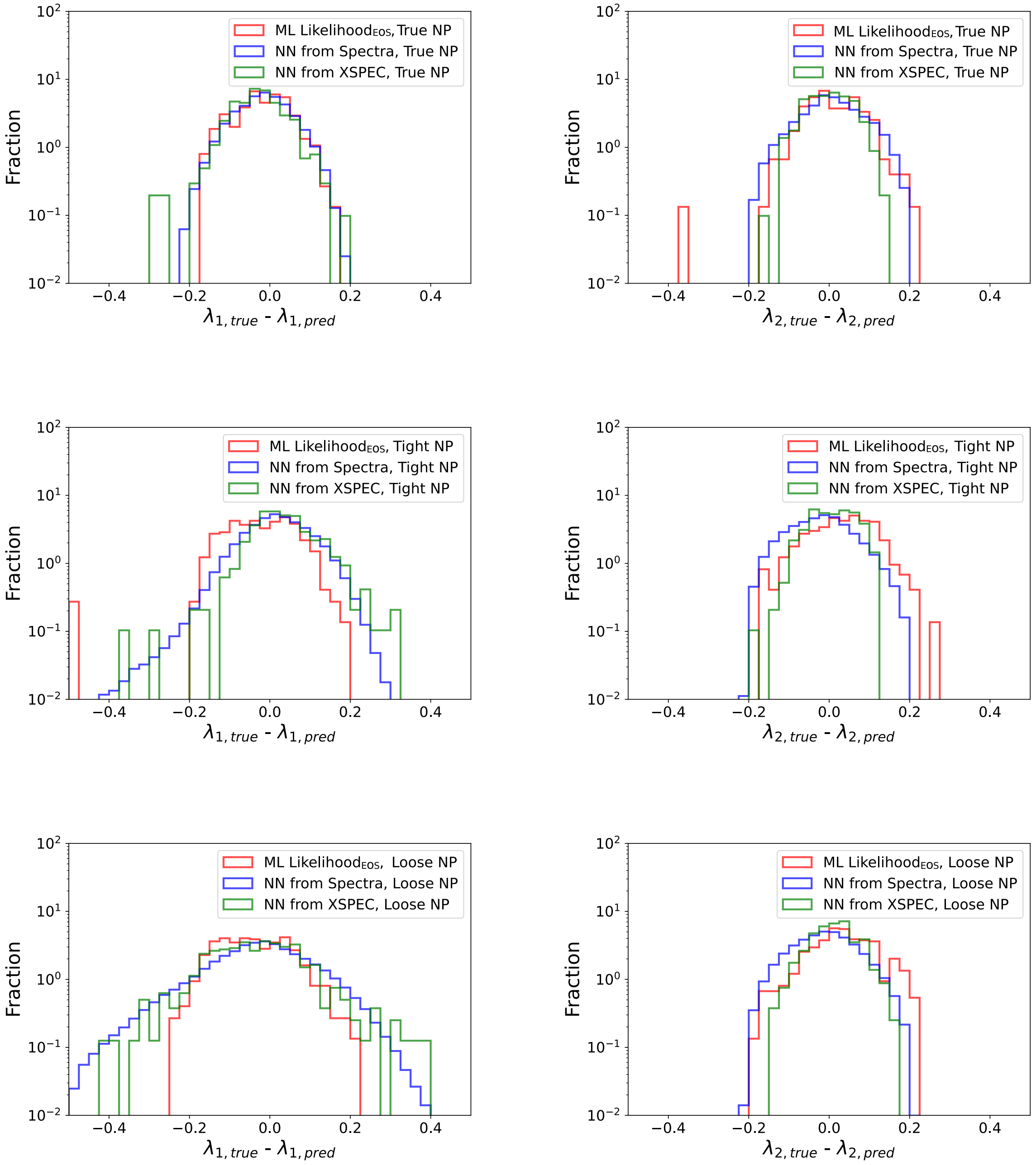

FIG. 8: Performance of our estimation of neutron star EOS parameters λ_1 (left) and λ_2 (right) using an approximate likelihood that incorporates a neural network, ML-Likelihood$_{\mathrm{EOS}}$, in comparison to the performance of a spectra-to-EOS regression network and network which regresses EOS parameters from M, R values estimated by XSPEC, both from Ref. [36]. Shown are the residual distributions, the difference between the true and predicted values, under three scenarios of nuisance parameter uncertainties. See Table III for quantitative analysis. In the "true" case, the NPs are fixed to their true values; in the "tight" and "loose" cases, they are drawn from narrow or wide priors, respectively; see text for details

Once released, this framework will serve as a convenient simulation tool that can be used by the larger machine learning research community to generate neutron star samples without the expert knowledge required to run XSPEC or solve the TOV equations, and spur further innovation, such as in simulator-based inference techniques that can handle

TABLE III: Performance of our estimation of neutron star EOS parameters λ_1 (left) and λ_2 (right) using an approximate likelihood that incorporates a neural network, ML-Likelihood$_{\mathrm{EOS}}$, in comparison to the performance of a spectra-to-EOS regression network and network which regresses EOS parameters from M, R values estimated by XSPEC, both from Ref. [36]. Shown are the mean (μ) and standard deviation (σ) of the residual distributions under three scenarios of nuisance parameter uncertainties. See Fig 8 for distributions. In the "true" case, the NPs are fixed to their true values; in the "tight" and "loose" cases, they are drawn from narrow or wide priors, respectively; see text for details

		λ_1		λ_2		Combined
Method	Nuis. Params.	μ	σ	μ	σ	σ
ML-Likelihood$_{\mathrm{EOS}}$	True	-0.02	0.066	0.01	0.070	0.096
NN(Spectra)	True	-0.02	0.066	0.01	0.075	0.099
NN(M, R via XSPEC)	True	-0.03	0.065	0.01	0.055	0.085
ML-Likelihood$_{\mathrm{EOS}}$	Tight	-0.02	0.078	0.03	0.081	0.112
NN(Spectra)	Tight	0.02	0.085	-0.02	0.077	0.115
NN(M, R via XSPEC)	Tight	-0.03	0.081	0.01	0.056	0.098
ML-Likelihood$_{\mathrm{EOS}}$	Loose	-0.04	0.089	0.03	0.081	0.120
NN(Spectra)	Loose	-0.03	0.131	-0.01	0.078	0.152
NN(M, R via XSPEC)	Loose	-0.03	0.123	0.01	0.058	0.136

a large number of nuisance parameters, or to setup ML challenges where participants are given access to the simulator.

Appendix C: Conclusion

In this work, we demonstrate an alternative approach to deducing neutron star mass, radius, and EOS from simulated stellar X-ray spectra using machine learning. The alternative approach employs a technique novel to neutron star astrophysics, machine-learning-derived likelihoods, in which intractable elements of the calculation are replaced with machine-learned functions. This allows for an approximate likelihood calculation with increased interpretability. Our forward neural network model allows us to predict the X-ray spectra given a value of nuisance parameters, the equation of state parameters, and the neutron star mass. Forward modeling of this kind demonstrates how these networks are interpretable, testable, and often with the assumptions being explicit rather than implicit when compared to inverse models [40, 57]. The ML-Likelihood$_{\mathrm{EOS}}$ model outperforms our previous best-performing regression model, demonstrating the power of such a technique for point estimation. Note that it additionally provides access to the full likelihood, which can inform uncertainty estimation.

While the studies shown in this work demonstrate the power of the method to provide a tractable likelihood for stellar spectra generated from a specific model, the natural ability of neural networks to interpolate allows for the treatment of inherent model uncertainties, by training on samples with mixed or varying models [27].

The machine learning-driven calculations lend themselves to various future directions. In terms of training data, EOS models with different parametrization can be tested as well as using a different response matrix (e.g. from NuStar, NICER, or XMM-Newton as described in [58]) in XSPEC to generate simulated X-ray spectra. In addition, this technique can be rapidly applied to the analysis of X-ray spectra from recent observations. The networks developed for this work will become a convenient tool for fast simulation of neutron star spectra which can be run without the expert knowledge required to run XSPEC or to solve the TOV equations.

Appendix D: Acknowledgements

DW and AG are supported by The Department of Energy Office of Science grant DE-SC0009920, and AG is also supported under contract DE-AC02-05CH11231. LL was supported by NSF Grant No. 2012857 to the University of California at San Diego. AWS was supported by NSF AST 22-06322, PHY 21-16686, and the Department of Energy Office of Nuclear Physics. DF and FW are supported by the National Science Foundation (USA) under Grant No. PHY-2012152. The authors are grateful to C. O. Heinke and W. C. G. Ho for their assistance in understanding and navigating the XSPEC software, and to Kyle Cranmer for conceptual contributions.

Appendix A: Hyperparameter Optimization

This section describes the hyperparameter optimization methods for model f, which generates simulated X-ray spectra from mass, radius, and the nuisance parameters ν.

The network has two input branches, the first to process the mass and radius values and the second to process ν. The output of the two branches is combined together and processed through the remaining layers to predict the X-ray spectra. In both the input branches as well as the secondary portion of the network, we trained models with random combinations of the following parameters:

1. Number of layers: one to fifteen,
2. Number of hidden nodes: 64 to 2048,
3. Dropout: 0 to 1,
4. Learning rate: 0.0001 to 0.005,
5. Activation functions,
6. Skip connections,
7. Loss functions: mean squared error (MSE), mean absolute percentage error (MAPE), and the Huber loss function,
8. Data scalers for both the input and the output: zero-mean, log, min-max, and standard scaler. The scalers preprocess the data before it enters the network, meaning scaled input values (mass, radius, and ν) with a scaled output (spectra) are used for training. After training, the scaling procedure is inverted for the predicted spectra to produce meaningful values.

The hyperparameters chosen were the combination that had the best performance on the validation data set, described in Section V.

[1] L. Tolos and L. Fabbietti, Prog. Part. Nucl. Phys. **112**, 103770 (2020), 2002.09223.

[2] J. J. Li, A. Sedrakian, and F. Weber, Physics Letters B **783**, 234 (2018).

[3] W. M. Spinella and F. Weber, Astronomische Nachrichten **340**, 145 (2019).

[4] G. Malfatti, M. G. Orsaria, I. F. Ranea-Sandoval, G. A. Contrera, and F. Weber, Phys. Rev. D **102**, 063008 (2020), URL https://link.aps.org/doi/10.1103/PhysRevD.102.063008.

[5] A. Sedrakian, J. J. Li, and F. Weber, Progress in Particle and Nuclear Physics p. 104041 (2023), ISSN 0146-6410, URL https://www.sciencedirect.com/science/article/pii/S0146641023000224.

[6] C. Alcock, E. Farhi, and A. Olinto, Astrophys. J. **310**, 261 (1986).

[7] M. Orsaria, H. Rodrigues, F. Weber, and G. Contrera, Physical Review C **89**, 015806 (2014).

[8] M. Alford, Annual Review of Nuclear and Particle Science **51**, 131 (2001), https://doi.org/10.1146/annurev.nucl.51.101701.132449, URL https://doi.org/10.1146/annurev.nucl.51.101701.132449.

[9] M. G. Alford, A. Schmitt, K. Rajagopal, and T. Schäfer, Rev. Mod. Phys. **80**, 1455 (2008), 0709.4635.

[10] J. Zdunik and P. Haensel, Astronomy & Astrophysics **551**, A61 (2013).

[11] G. Baym, Physical Review Letters **30**, 1340 (1973).

[12] D. Kaplan and A. Nelson, Physics Letters B **175**, 57 (1986), ISSN 0370-2693, URL https://www.sciencedirect.com/science/article/pii/037026938690331X.

[13] N. K. Glendenning and J. Schaffner-Bielich, Phys. Rev. C **60**, 025803 (1999), URL https://link.aps.org/doi/10.1103/PhysRevC.60.025803.

[14] P. J. Ellis, R. Knorren, and M. Prakash, Physics Letters B **349**, 11 (1995).

[15] A. Ramos, J. Schaffner-Bielich, and J. Wambach, Kaon Condensation in Neutron Stars (Springer Berlin Heidelberg, Berlin, Heidelberg, 2001), pp. 175–202, ISBN 978-3-540-44578-4, URL https://doi.org/10.1007/3-540-44578-1_6.

[16] A. W. Steiner, M. Hempel, and T. Fischer, The Astrophysical Journal **774**, 17 (2013).

[17] K. Hotokezaka, K. Kyutoku, H. Okawa, M. Shibata, and K. Kiuchi, Physical Review D **83**, 124008 (2011).

[18] R. E. Rutledge, L. Bildsten, E. F. Brown, G. G. Pavlov, and V. E. Zavlin, Astrophys. J. **514**, 945 (1999), astro-ph/9810288.

[19] C. O. Heinke, G. B. Rybicki, R. Narayan, and J. E. Grindlay, Astrophys. J. **644**, 1090 (2006), astro-ph/0506563.

[20] J. M. Lattimer and M. Prakash, Astrophys. J. **550**, 426 (2001), URL https://doi.org/10.1086/319702.

[21] L. Lindblom, Phys. Rev. D **82**, 103011 (2010), URL https://link.aps.org/doi/10.1103/PhysRevD.82.103011.

[22] A. W. Steiner, J. M. Lattimer, and E. F. Brown, Astrophys. J. **722**, 33 (2010), 1005.0811, URL http://doi.org/10.1088/0004-637X/722/1/33.

[23] L. Lindblom and N. M. Indik, Phys. Rev. D **89**, 064003 (2014), [Erratum: Phys.Rev.D 93, 129903 (2016)], 1310.0803.

[24] L. Lindblom and N. Indik, Physical Review D **86** (2013).

[25] P. Baldi, Deep Learning in Science (Cambridge University Press, Cambridge, UK, 2021).

[26] P. Baldi, P. Sadowski, and D. Whiteson, Nature Communications **5** (2014).

[27] A. Ghosh, B. Nachman, and D. Whiteson, Phys. Rev. D **104**, 056026 (2021), 2105.08742.

[28] P. Baldi, K. Cranmer, T. Faucett, P. Sadowski, and D. Whiteson, Eur. Phys. J. C **76**, 235 (2016), 1601.07913.

[29] D. Guest, J. Collado, P. Baldi, S.-C. Hsu, G. Urban, and D. Whiteson, Phys. Rev. D **94**, 112002 (2016), 1607.08633.

[30] K. Cranmer, J. Brehmer, and G. Louppe, Proceedings of the National Academy of Sciences **117**, 30055 (2020), URL https://doi.org/10.1073%2Fpnas.1912789117.

[31] Y. Fujimoto, K. Fukushima, and K. Murase, Phys. Rev. D **101**, 054016 (2020), 1903.03400.

[32] Y. Fujimoto, K. Fukushima, and K. Murase, JHEP **03**, 273 (2021), 2101.08156.

[33] F. Morawski and M. Bejger, Astron. Astrophys. **642**, A78 (2020), 2006.07194.

[34] S. Soma, L. Wang, S. Shi, H. Stöcker, and K. Zhou, Journal of Cosmology and Astroparticle Physics **2022**, 071 (2022).

[35] M. Ferreira and C. Providência (2019), 1910.05554.

[36] D. Farrell et al. (2022).

[37] G. Papamakarios, Neural density estimation and likelihood-free inference (2019), URL https://arxiv.org/abs/1910.13233.

[38] R. W. Romani, D. Kandel, A. V. Filippenko, T. G. Brink, and W. Zheng, The Astrophysical Journal Letters **934**, L18 (2022).

[39] K. Hebeler, J. Lattimer, C. J. Pethick, and A. Schwenk, The Astrophysical Journal **773**, 11 (2013).

[40] A. W. Steiner, C. O. Heinke, S. Bogdanov, C. Li, W. C. G. Ho, A. Bahramian, and S. Han, Mon. Not. Roy. Astron. Soc. **476**, 421 (2018), 1709.05013, URL https://doi.org/10.1093/mnras/sty215.

[41] C. O. Heinke, J. E. Grindlay, P. M. Lugger, H. N. Cohn, P. D. Edmonds, D. A. Lloyd, and A. M. Cool, The Astrophysical Journal **598**, 501 (2003), URL https://doi.org/10.1086/378885.

[42] S. Bogdanov, C. O. Heinke, F. Özel, and T. Güver, Astrophys. J. **831**, 184 (2016), 1603.01630.

[43] S. Campana, M. Colpi, S. Mereghetti, L. Stella, and M. Tavani, The Astronomy and Astrophysics Review **8**, 279 (1998).

[44] A. Y. Potekhin, Physics-Uspekhi **57**, 735 (2014).

[45] P. Slane, D. Helfand, and S. Murray, Astrophysical Journal - ASTROPHYS J **571** (2002).

[46] R. Wijnands, N. Degenaar, and D. Page, Journal of Astrophysics and Astronomy **38**, 1 (2017).

[47] H. Sun, Y. Li, B.-B. Zhang, B. Zhang, F. Bauer, Y. Xue, and W. Yuan, The Astrophysical Journal **886**, 129 (2019).

[48] K. A. Arnaud, in Astronomical Data Analysis Software and Systems V, edited by G. H. Jacoby and J. Barnes (1996), vol. 101 of Astronomical Society of the Pacific Conference Series, p. 17.

[49] P. Baldi, K. Bauer, C. Eng, P. Sadowski, and D. Whiteson, Physical Review D **93**, 094034 (2016).

[50] M. J. Fenton, A. Shmakov, T.-W. Ho, S.-C. Hsu, D. Whiteson, and P. Baldi, Physical Review D **105**, 112008 (2022), also arXiv:2010.09206.

[51] S. Typel, G. Röpke, T. Klähn, D. Blaschke, and H. H. Wolter, Phys. Rev. C **81**, 015803 (2010), URL https://link.aps.org/doi/10.1103/PhysRevC.81.015803.

[52] M. Oertel, M. Hempel, T. Klähn, and S. Typel, Rev. Mod. Phys. **89**, 015007 (2017), URL https://link.aps.org/doi/10.1103/RevModPhys.89.015007.

[53] L. Lindblom, Phys. Rev. D **97**, 123019 (2018), URL https://link.aps.org/doi/10.1103/PhysRevD.97.123019.

[54] C. O. Heinke, G. B. Rybicki, R. Narayan, and J. E. Grindlay, The Astrophysical Journal **644**, 1090 (2006), URL https://doi.org/10.1086/503701.

[55] J. M. Lattimer and A. W. Steiner, Astrophys. J. **784**, 123 (2014), 1305.3242, URL http://doi.org/10.1088/0004-637X/784/2/123.

[56] A. Ghosh, X. Ju, B. Nachman, and A. Siodmok, Phys. Rev. D **106**, 096020 (2022), 2203.12660.

[57] F. Özel and D. Psaltis, Astrophys. J. **810**, 135 (2015), 1505.05156.

[58] R. Ludlam, J. Miller, Z. Arzoumanian, P. Bult, E. Cackett, D. Chakrabarty, T. Dauser, T. Enoto, A. Fabian, J. García, et al., The Astrophysical Journal Letters **858**, L5 (2018).