Star Log-extended eMulation: A method for efficient computation of the Tolman-Oppenheimer-Volkoff equations

Sudhanva Lalit [1,*] Alexandra C. Semposki [2,†] and Joshua M. Maldonado [2,‡]
[1]*Facility for Rare Isotope Beams, Michigan State University, East Lansing, Michigan 48824, USA*
[2]*Department of Physics and Astronomy and Institute of Nuclear and Particle Physics, Ohio University, Athens, Ohio 45701, USA*

(Received 22 November 2024; accepted 9 September 2025; published 10 October 2025)

We emulate the Tolman-Oppenheimer-Volkoff (TOV) equations, including tidal deformability, for neutron stars using a method based upon the Dynamic Mode Decomposition. This method, which we call Star Log-extended eMulation (SLM), utilizes the underlying logarithmic behavior of the differential equations to enable accurate emulation of the nonlinear system. We show predictions for well-known equations of state (EOSs) with fixed parameters using the SLM, accurately recreating high-fidelity results while achieving a computational speedup of $\approx 2.4 \times 10^4$. We test our parametric SLM method for a two-parameter quarkyonic EOS against high-fidelity RK4 TOV calculations and find a computational speedup of $\approx 7.0 \times 10^4$. Hence, SLM is an efficient emulator for the numerous TOV evaluations required by multimessenger astrophysical frameworks that infer constraints on the EOS. The ability of the SLM algorithm to learn a mapping between parameters of the EOS and subsequent neutron star properties also opens up potential extensions for assisting in computationally prohibitive uncertainty quantification for any type of EOS. The source code for the methods employed in this work is openly available in a public GitHub repository for community modification and use.

DOI: 10.1103/5p3h-b8rf

I. INTRODUCTION

The advent of multimessenger astronomy, accelerated by LIGO's gravitational wave measurements [1–3] and NICER's pulsar observations [4–8], has coincided perfectly with advances in Bayesian uncertainty quantification (UQ) in nuclear physics [9,10] and the Facility for Rare Isotope Beams (FRIB) era of experiment [11]. These three complementary approaches enable a refined understanding of the strong interaction, which governs matter from finite nuclei to dense, stellar compact objects, e.g., neutron stars. These intriguing astrophysical laboratories provide us with critical insights on extreme matter not found terrestrially. However, the lack of direct probes into the interior composition of neutron star cores indicates we must rely on the combination of information from neutron-star observations and binary mergers, nuclear experiment [12,13], and the construction of state-of-the-art nuclear potentials to infer the equation of state (EOS) of the star [14–16]. This thermodynamic description, defined in terms of the state variables, i.e., pressure, number density, and energy density, is then propagated through the Tolman-Oppenheimer-Volkoff (TOV) equations [17,18] to determine the corresponding mass and radius of the neutron star. Future gravitational wave (GW) measurements, as well as data from NICER and Cosmic Explorer, will provide more constraints on this mass-radius relationship, including other important properties, such as the tidal deformability [19–23], which will, in turn, allow us to place tighter constraints on the EOS.

This multimessenger era of astrophysics has led to the need for more efficient calculations of the TOV equations due to their use in large-scale frameworks, e.g., LIGO's GW inference framework [1,24,25], where thousands of solutions can be required for reliable UQ of the mass-radius posterior. The precision era of nuclear physics [16,26–30] has also led to the use of smaller-scale Bayesian frameworks that require sampling of the EOS and the computation of posteriors for neutron star properties [16,29,31,32], hence necessitating numerous solutions to the TOV equations. To perform these calculations efficiently, the computational burden of solving the TOV equations, especially when including tidal deformability, must be overcome. An approach already employed in nuclear physics is the design and use of sophisticated emulators to reduce computational time and resources. Recent efforts in constructing model-intrusive, projection-based emulators have produced remarkable results in nuclear scattering [33–37], and model-extrusive emulators such as Gaussian processes have been used successfully in nuclear astrophysics [29,38].

In this work, we present an emulation strategy based upon the Dynamic Mode Decomposition (DMD) method used in analyses of time dynamics [39–42], supplemented by machine learning techniques. To differentiate this from standard DMD, we call our method Star Log-extended eMulation (SLM).

*Contact author: lalit@frib.msu.edu
†Contact author: as727414@ohio.edu
‡Contact author: jm998521@ohio.edu

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

Currently available DMD libraries [43] are incapable of handling the nonlinearity of the TOV equations; hence, this work establishes an emulator that overcomes this hurdle. We apply two SLM emulators to the TOV equations, including tidal deformability, for the first time.

This paper is organized as follows. In Sec. II, we review the TOV equations, including tidal deformability, and scale them for use with our high-fidelity RK4 solver. In Sec. III, we give a general introduction to DMDs, which will be the foundation on which we build our SLM routines. We then detail the formalism, implementation, and results of our initial SLM routine in Sec. IV, which we apply to five tabular EOSs. We use these to benchmark our emulator against their corresponding high-fidelity (HF) solutions. We then employ a parametric version of our emulator, called parametric SLM (pSLM), which we introduce and apply to a two-parameter quarkyonic EOS in Sec. V. This emulator learns a mapping between EOS parameters and previously mentioned neutron star properties, a necessary feature for the Bayesian calibration of other phenomenological EOSs with freely varying parameters. We compare our SLM and pSLM results to low-fidelity solvers, i.e., RK2 and forward Euler methods, in Sec. VI. Finally, we summarize our results and point to future applications of our work in Sec. VII.

Our emulators, along with the code to produce the results in this paper, are publicly available for community use in our GitHub repository [44].

II. NEUTRON STAR PROPERTIES

Macroscopic properties of neutron stars, such as mass, radius, and tidal deformability are determined by solving the TOV equations simultaneously with appropriate differential equations corresponding to each additional desired quantity. The TOV equations, along with the tidal deformability equation, are given by [17,20–22,45]

$$\frac{dP}{dr} = -\frac{G}{c^2}[\epsilon(r) + P(r)]\frac{m(r) + 4\pi r^3 P(r)/c^2}{r[r - 2Gm(r)/c^2]}, \tag{1}$$

$$\frac{dm}{dr} = 4\pi r^2 \frac{\epsilon(r)}{c^2}, \tag{2}$$

$$\frac{dy}{dr} = -\frac{y(r)^2}{r} - \frac{F(r)y(r)}{r} - \frac{Q(r)}{r}, \tag{3}$$

where

$$F(r) = \frac{1 - 4\pi G r^2[\epsilon(r) - P(r)]/c^4}{1 - 2\frac{Gm(r)}{rc^2}} \tag{4}$$

and

$$Q(r) = \frac{4\pi G r^2/c^4}{1 - \frac{2Gm(r)}{rc^2}} \times \left(5\epsilon(r) + 9p(r) + \frac{\epsilon(r) + p(r)}{c_s(r)^2}c^2 - \frac{6c^4}{4\pi r^2 G}\right) - 4\left(\frac{G[m(r)/(rc^2) + 4\pi r^2 p(r)/c^4]}{1 - 2Gm(r)/(rc^2)}\right)^2. \tag{5}$$

The dimensionless versions of these equations can be obtained by scaling them using the parameters in Ref. [46]. We scale the differential equation for tidal deformability in a similar fashion (see Appendix A for details). The resulting equations are

$$\frac{dp}{dx} = -\frac{1}{2}\frac{[\varepsilon(x) + p(x)][m(x) + 3x^3 p(x)]}{x^2(1 - m(x)/x)}, \tag{6}$$

$$\frac{dm}{dx} = 3x^2\varepsilon(x), \tag{7}$$

$$\frac{dy}{dx} = -\frac{y(x)^2}{x} - \frac{F(x)y(x)}{x} - \frac{Q(x)}{x}, \tag{8}$$

with

$$F(x) = \left[1 - \tfrac{3}{2}x^2(\varepsilon(x) - p(x))\right][1 - m(x)/x]^{-1} \tag{9}$$

and

$$Q(x) = \frac{\frac{3}{2}x^2}{\left(1 - \frac{m(x)}{x}\right)}\left[5\varepsilon(x) + 9p(x) + \frac{\varepsilon(x) + p(x)}{c_s^2(x)}c^2 - \frac{4}{x^2}\right] - \left(\frac{m(x)/x + 3x^2 p(x)}{\left(1 - \frac{m(x)}{x}\right)}\right)^2, \tag{10}$$

where $p(x)$, $\varepsilon(x)$, $m(x)$, and $c_s^2(x)$ are the scaled dimensionless pressure, energy density, mass, and squared speed of sound functions, respectively, while $y(x)$ is the scaled solution to the tidal equation. The dense matter EOS is an essential input to this system through $p(x)$ and $\varepsilon(x)$.

We solve Eqs. (6)–(8) as initial value problems for a given central pressure p_c, mass $m = 0$, and $y = 2.0$ (from Ref. [22]) at the center of the star. The radius and mass of the star are determined by the condition $p(x) \to 0$ at the edge of the star. In this work, we have employed an HF fourth-order Runge-Kutta (RK4) solver. We also construct the dimensionless coefficient $k_2(R)$, known as the Love number [22], which is computed at the radius R of the star. $k_2(R)$ is computed as

$$\begin{aligned} k_2(R) = {} & \frac{8\beta^5}{5}(1 - 2\beta)^2[2 + 2\beta(y_R - 1) - y_R] \\ & \times \{2\beta(6 - 3y_R + 3\beta(5y_R - 8)) \\ & + 4\beta^3[13 - 11y_R + \beta(3y_R - 2) + 2\beta^2(1 + y_R)] \\ & + 3(1 - 2\beta)^2[2 - y_R + 2\beta(y_R - 1)]\log(1 - 2\beta)\}^{-1}, \end{aligned} \tag{11}$$

where $\beta = GM(R)/R$ is the compactness of the star and $y_R = y(r = R)$. Here, we use the unscaled solutions of the TOV equations.

We have chosen input EOSs to the TOV equations that can obtain a maximum mass of greater than $2M_\odot$, possess different sets of underlying parameters, and enable us to test our algorithm on a variety of masses and radii, as well as various structures of the Love number curve $k_2(R)$. Our HF solutions for their maximum masses and corresponding radii are presented in Table I and are consistent with the results in the compOSE database [47] and from the original publications [48]. In Secs. IV B and V B, we compare these HF solutions to our emulated results.

III. DYNAMIC MODE DECOMPOSITION

DMD [39,40], a data-driven technique, is a model-extrusive emulator and has been successful in emulating

TABLE I. Comparisons of the HF and emulated maximum mass and corresponding radii for various EOSs [48,59–63] selected from the compOSE database [47], as well as the same quantities computed for the quarkyonic EOS [66] with various choices of parameters Λ and κ are shown. The tabular EOSs are emulated using the SLM algorithm (see Fig. 2), whereas the quarkyonic EOS is emulated through the pSLM algorithm (see Fig. 3). The remarkable efficiency achieved is seen in the large difference between the time spent computing the HF and emulated solutions (last two columns), leading to an average speedup of 4.7×10^4.

	HF		SLM		Rel. error (%)		Time (s)	
EOS	Max. mass ($M_\odot$)	Radius (km)	Max. mass ($M_\odot$)	Radius (km)	Max. mass	Radius	HF	SLM
				Tabular EOS				
SLy4	2.067	10.033	2.066	10.031	0.04	0.01	10.711	3.841×10^{-4}
APR	2.193	9.979	2.193	9.981	0.01	0.02	10.856	4.580×10^{-4}
FSU Garnet	2.066	11.601	2.066	11.598	0.00	0.02	10.934	4.520×10^{-4}
BL	2.083	10.341	2.083	10.345	0.01	0.04	10.767	5.159×10^{-4}
DS-CMF-5	2.023	11.802	2.032	11.597	0.46	1.73	10.827	4.659×10^{-4}
				Quarkyonic EOS				
$\Lambda = 300.00, \kappa = 0.26$	2.666	12.614	2.658	12.642	0.30	0.22	11.718	1.600×10^{-4}
$\Lambda = 352.63, \kappa = 0.12$	2.329	10.671	2.322	10.632	0.29	0.36	11.421	1.643×10^{-4}
$\Lambda = 384.21, \kappa = 0.23$	2.476	11.517	2.454	11.521	0.88	0.04	11.323	1.559×10^{-4}
$\Lambda = 436.84, \kappa = 0.14$	2.887	14.532	2.891	14.580	0.14	0.33	11.238	1.540×10^{-4}
$\Lambda = 500.00, \kappa = 0.28$	2.224	10.177	2.225	10.172	0.06	0.05	11.225	1.788×10^{-4}

dynamical systems in the areas of fluid dynamics, neuroscience, and financial systems [49–51]. Fundamentally, the method can be viewed as a blend of spatial dimensionality reduction techniques, such as proper orthogonal decomposition and Fourier transforms in the time domain. In this section, we follow Ref. [40] closely.

Consider a set of nonautonomous coupled differential equations

$$\frac{\partial \vec{\xi}}{\partial t} = \mathcal{F}(\vec{\xi}(t), t), \quad \vec{\xi}(t) \in \mathbb{R}^n, \tag{12}$$

where $\mathcal{F}$ is a nonlinear operator that describes the dynamics of the governing equations of interest $\vec{\xi}(t)$, and where $\vec{\xi}(t)$ is the system's state at some given time t. Finding the exact form of $\mathcal{F}$ can be prohibitively difficult, so to mitigate this issue we employ the approximately locally linear characteristic of these systems, discretize the phase space in terms of equally spaced time steps Δt, and write

$$\frac{d\vec{\xi}}{dt} = \mathcal{A}\vec{\xi}, \tag{13}$$

where $\mathcal{A}$ is the locally linear approximation of the dynamics originally described by $\mathcal{F}$. Hence, it locally governs the evolution of the system, even if the true dynamics are nonlinear. However, in practice we need to discretize this in terms of time steps Δt, so we do this by writing a discretized version of $\mathcal{A}$: $\mathbf{A} = \exp(\mathcal{A}\Delta t)$. Equation (13) hence becomes

$$\frac{d\vec{\xi}}{dt} = \mathbf{A}\vec{\xi}. \tag{14}$$

To determine $\mathbf{A}$, we will construct matrices of snapshots of our system at different time steps. These are generally denoted as

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \xi_1 & \xi_2, & \ldots, & \xi_{m-1} \\ | & | & & | \end{bmatrix},$$
$$\mathbf{X}' = \begin{bmatrix} | & | & & | \\ \xi_2 & \xi_3, & \ldots, & \xi_m \\ | & | & & | \end{bmatrix}. \tag{15}$$

The locally linear approximation is devised from here by writing $\mathbf{X}' \approx \mathbf{A}\mathbf{X}$. To find the best-fit value of $\mathbf{A}$, we can then write $\mathbf{A} = \mathbf{X}'\mathbf{X}^\dagger$ using the Moore-Penrose pseudoinverse. To save computational time and effort, and to ensure our results are numerically stable, we then find the reduced rank evolution operator, which we denote as $\mathbf{A}_s$; this is done by projecting the snapshot matrix $\mathbf{X}$ into a low-dimensional subspace via the Singular Value Decomposition (SVD),

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*, \tag{16}$$

where $\mathbf{U} \in \mathbb{C}^{m\times m}$, $\mathbf{\Sigma} \in \mathbb{C}^{m\times n}$, and $\mathbf{V} \in \mathbb{C}^{n\times n}$. We pick s singular values (also called modes) that approximate $\mathbf{A}$ to a desired accuracy. This can be achieved by

$$s = \text{argmin}_j \frac{\Sigma_{i=1}^{i=j}\sigma_i}{\Sigma_{i=1}^{i=k}\sigma_i}, \eta, \tag{17}$$

where σ are the j singular values chosen from the SVD, k is the total number of singular values, and η is a value between 0 and 1 that we must choose for our problem [42].

Hence, we now have $\mathbf{U}_s \in \mathbb{C}^{m\times s}$, $\mathbf{\Sigma}_s \in \mathbb{C}^{s\times s}$, and $\mathbf{V}_s \in \mathbb{C}^{n\times s}$. We use these reduced matrices to express $\mathbf{A}_s$ via the pseudoinverse ($\mathbf{X}^\dagger$) as

$$\mathbf{X}^\dagger \approx \mathbf{V}_s\mathbf{\Sigma}_s^{-1}\mathbf{U}_s^* \tag{18}$$

to obtain

$$\mathbf{A}_s = \mathbf{X}'\mathbf{V}_s\mathbf{\Sigma}_s^{-1}\mathbf{U}_s^*. \tag{19}$$

Now, to derive $\mathbf{A}_s$ in the projected low-dimensional subspace, we take the projected matrix using

$$\mathbf{A}_s = \mathbf{U}_s^* \mathbf{A}_s \mathbf{U}_s = \mathbf{U}_s^* \mathbf{X}_s' \mathbf{V}_s \mathbf{\Sigma}_s^{-1}. \tag{20}$$

From here, we can compute the eigendecomposition of $\mathbf{A}_s$ using

$$\mathbf{A}_s \mathbf{W}_s = \mathbf{W}_s \mathbf{\Lambda}_s, \tag{21}$$

where $\mathbf{\Lambda}_s$ is the diagonal matrix of the eigenvalues of $\mathbf{A}_s$ and $\mathbf{W}_s$ is the matrix of corresponding eigenvectors.

We can then construct the eigenvectors $\mathbf{\Phi}$ of the solutions $\vec{\xi}(t)$ through the "projected DMD modes," which are found through $\mathbf{\Phi} = \mathbf{U}_s \mathbf{W}_s$. We also define $\omega_k = \ln(\lambda_k)/\Delta t$ and obtain $\vec{b}$ from the pseudoinverse $\vec{b} = \mathbf{\Phi}^\dagger \vec{\xi}$. Finally, we can construct the equation of solutions to our system, which are

$$\vec{\xi}(t) = \sum_{k=1}^{r} \vec{\phi}_k \exp(\omega_k t) b_k \equiv \mathbf{\Phi} \exp(\mathbf{\Omega} t) \vec{b} \tag{22}$$

for all future times. This equation-free structure makes DMDs very efficient; they are constrained only by the size of the rank s chosen. The emulators in the following two sections are built upon this standard DMD but are designed to obtain better results for our nonlinear TOV system.

IV. STAR LOG-EXTENDED EMULATION

Here, we use the foundational aspects of DMDs to build our emulator for the TOV equations. We discuss the formalism of this emulator in Sec. IV A and move to our results for a selection of tabular EOSs in Sec. IV B.

A. Formalism

DMD-based emulators are generally applicable to problems that have polynomial nonlinearities or can be expressed as an eigenvalue problem [33,34,37,52] and are used extensively for the emulation of various time-dependent systems [42,53–55]. Recent works applied DMDs to nonlinear problems [42] including a time-independent scenario [56]; however, the DMD algorithms presented there suffer from the inability to capture the more complex nonlinear dynamics of the coupled TOV system. To overcome this limitation, we extend the DMD method to one that can handle the TOV equations by building a logarithmic structure into the DMD modes.

Thus, in our case, we solve a nonautonomous system of equations, which can be written as

$$\frac{d\vec{\xi}}{dr} = \mathcal{F}(\vec{\xi}(\vec{\alpha}, r), r), \tag{23}$$

where $\mathcal{F}$ is a nonlinear operator that describes the governing equation of interest by discretizing the phase space, i.e., taking snapshots at discrete values of r. Here, r is the independent variable, while $\vec{\alpha}$ consists of other dependent variables such as ε. We therefore wish to find

$$\frac{d\vec{\xi}}{dr} = \mathbf{A}\vec{\xi}, \tag{24}$$

where the linear operator $\mathbf{A}$ approximates the dynamics of the system as in Sec. III. This operator is learned in a data-driven fashion.

ALGORITHM 1. SLM. We closely follow Refs. [40,42] for the foundation of this algorithm. Italicized portions of the algorithm are direct extensions to Refs. [40,42].

1: Solve Eq. (23) and collect snapshots $[\vec{\xi}_i]_{i=0}^m$, *where i are the data indices.*
2: *Compute the logarithm of these snapshots as* $[\log \vec{\xi}(r_i)]_{i=0}^m$.
3: *Extend the snapshots by adding the Hadamard product of the arrays, i.e.,* $\prod_{i,j} \log \vec{\xi}_i \log \vec{\xi}_j$.
4: Organize the snapshots in $\mathbf{X}$ and $\mathbf{X}'$ data matrices.
5: Perform SVD of $\mathbf{X}$: $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathbf{T}}$.
6: Keep s modes to desired accuracy and compute the reduced Koopman operator $\mathbf{A}_s \equiv \mathbf{U}_s^T \mathbf{A} \mathbf{U}_s = \mathbf{U}_s^T \mathbf{X}' \mathbf{V}_s \mathbf{\Sigma}_s^{-1}$.
7: Perform eigendecomposition of $\mathbf{A}_s$ to obtain the reduced eigenmodes: $\mathbf{A}_s \mathbf{W} = \mathbf{\Lambda} \mathbf{W}$.
8: Produce the full-state modes $\mathbf{\Phi} = \mathbf{U}_s \mathbf{W}$.
9: Reconstruct the coupled, *nonlinear* ODE solutions $\vec{\xi}(r)$ using the first n modes with Eq. (22) and *exponentiate the solutions*.

In the DMD procedure, we replace the independent variable t with the linear index ℓ of the data in the training set and include the radius r in the set of snapshots directly. Since the linear index is linearly spaced and monotonically increasing, this helps us to handle the nonlinearity in the spacing of the training set.

Our snapshot matrices $\mathbf{X}$ and $\mathbf{X}'$ now can contain either the (unscaled) training data from the solutions of the TOV equations, i.e., $m(r)$, $p(r)$, $y(r)$, and r, or the final curves corresponding to $M(R)$, radius R, central pressure $P_c(R)$, and $k_2(R)$. In the first case, the SLM routine will train on the profile curves from the various initial pressures used to generate the TOV results for a given EOS. In the second case, the SLM routine instead trains on the final curves that are obtained from finding the points on the profile curve that correspond to the radius R. Results for both of these cases are shown in the next section.

Once we choose which training data to use, we then take the logarithm of these snapshots, i.e., $\vec{z} = \log\{\vec{\xi}_i\}_{i=0}^m \in \mathbb{R}^{n \times m}$, where i indexes the m total snapshots. Thus, in the first case, $\vec{z}$ contains solutions corresponding to $\log(y(r))$, $\log(m(r))$, $\log(p(r))$, and $\log(r)$. We then employ the formalism of extended DMDs in Ref. [57], which is a machine-learning technique, to modify the snapshots further to make them suitable for our calculations, i.e., not only including the snapshot vectors themselves in the snapshot matrices $\mathbf{X}$ and $\mathbf{X}'$, but also taking the Hadamard product of the snapshot vectors. This approach significantly enhances reconstruction and prediction capabilities of the data.

The snapshots $\vec{z}$ and their Hadamard products are arranged in the matrices $\mathbf{X}$ and $\mathbf{X}' \in \mathbb{R}^{n \times m(m-1)}$, which are now defined as

$$\mathbf{X} = \begin{bmatrix} \vec{z}_1 & \cdots & \vec{z}_{m-1} & \vec{z}_1^2 & \vec{z}_1\vec{z}_2 & \cdots & \vec{z}_i\vec{z}_j & \cdots & \vec{z}_{m-1}^2 \end{bmatrix},$$
$$\mathbf{X}' = \begin{bmatrix} \vec{z}_2 & \cdots & \vec{z}_m & \vec{z}_2^2 & \vec{z}_2\vec{z}_3 & \cdots & \vec{z}_i\vec{z}_j & \cdots & \vec{z}_m^2 \end{bmatrix}. \tag{25}$$

We then follow the DMD procedure outlined in Sec. III, and included in Algorithm 1, using these snapshot matrices. For our use of SLM with the TOV equations, we chose

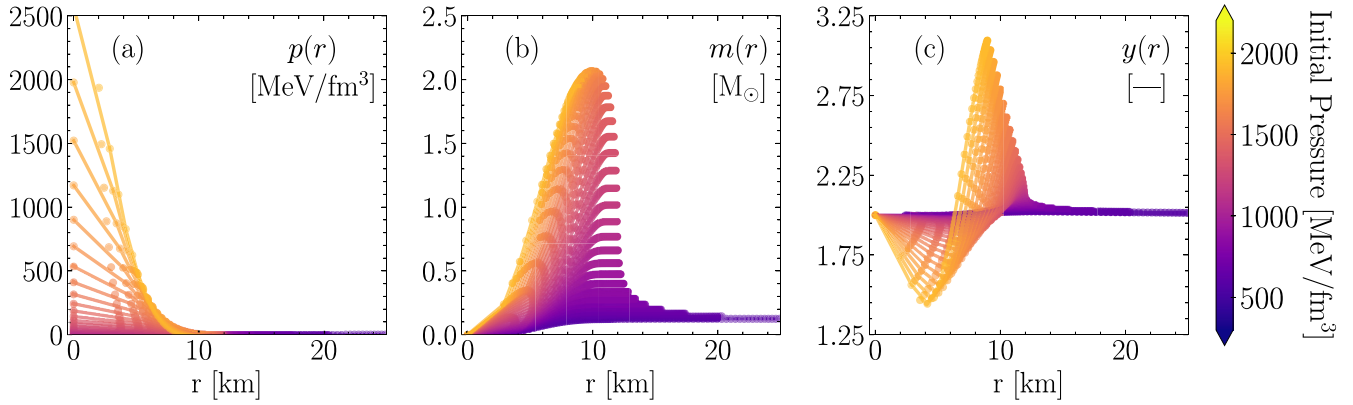


FIG. 1. High-fidelity solutions to the TOV and tidal deformability coupled differential equations (dotted curves) for the SLy4 tabular EOS [47] at various initial pressures, and the corresponding SLM predictions (solid curves). These results highlight the ability of the SLM routine to capture the underlying structure of the nonlinear differential equations.

$\eta = 0.9999$ in Eq. (17) to determine the number of modes for the SLM routine.

B. Results

As a test of our algorithm, we first emulate the solutions of the $m(r)$, $p(r)$, and $y(r)$ profile curves of a selected EOS. For this task, we utilize the well-known SLy4 EOS as input to the TOV equations for our HF calculations. Figure 1 shows the solutions to Eqs. (6)–(8) as functions of radius r and the output from the SLM algorithm for each quantity, respectively. The emulator is able to recover the solutions from the HF solver, allowing for accurate predictions between HF solutions. Hence, SLM only needs a few solutions of each quantity to reliably reconstruct the HF results [58].

To apply the SLM algorithm directly to the HF calculations of the M–R, P_c–R, and k_2–R curves, we choose five realistic equations of state that can produce a maximum mass of $\geqslant 2\,M_\odot$ [48,59–63]. These EOSs have different sets of underlying parameters. For instance, the APR is a nonrelativistic EOS employing the Urbana-Argonne potential, while FSU Garnet is a relativistic mean-field EOS. Figure 2 shows our SLM routine applied to these HF calculations using 50 total snapshots. The respective relative errors between the HF and emulated results were found to be $\leqslant 3 \times 10^{-3}$. These values are shown in the insets in Fig. 2. With an average speed of about 4.6×10^{-4} s, compared to the HF average speed of $\approx$11 s, the SLM routine is shown to be remarkably computationally efficient. This proof-of-principle calculation indicates that SLM can be used to accurately recover the structure of each curve through prediction between HF solutions corresponding to a given EOS.

Additionally, Table I shows the numerical results for the maximum masses and corresponding radii calculated from the HF RK4 solver for these EOSs, compared with the same quantities using the SLM algorithm, and the subsequent percent error between the HF and emulated calculations. All errors are $\leqslant$2% for the maximum mass and the corresponding radii.

V. PARAMETRIC SLM

We now turn to the parametric version of our SLM emulator, which we call parametric SLM. We discuss the modifications to our SLM algorithm for the parametric case in Sec. V A, and then show our implementation and results in Sec. V B.

A. Formalism

The pSLM method is based upon the reduced EigenPair Interpolation method from Ref. [42], using our modified snapshots (see Sec. IV A). However, instead of using a pointwise Lagrange interpolation as in Ref. [42], we use a recently developed technique based on the Greedy Recombination Interpolation Method for Banach spaces (B-GRIM) [64] to interpolate more accurately within the parameter space.

The pSLM method expands upon the general algorithm for SLM presented in the previous section, in order to determine the evolution of the system under parametric dependence. The governing equations in Eq. (23) may involve specific parameters $\mu_i \in \vec{\mu}$, $i = 1, 2, \ldots, n$ allowing them to be reformulated in a parametric form

$$\frac{\partial \vec{\xi}^{\,\vec{\mu}}}{\partial r} = \mathcal{F}(\vec{\xi}^{\,\vec{\mu}}(\vec{\alpha}, r; \vec{\mu}), r; \vec{\mu}), \tag{26}$$

where $\vec{\alpha}$ is a vector of other variables that $\vec{\xi}$ depends on.

After discretizing the phase space, this leads to a system of coupled parametric ordinary differential equations

$$\frac{d\vec{\xi}^{\,\vec{\mu}}}{dr} = \mathcal{A}\vec{\xi}^{\,\vec{\mu}}(\vec{\alpha}, r; \vec{\mu}). \tag{27}$$

The goal here is to replace the nonlinear operator $\mathcal{A}(\cdot)$ with a linear surrogate approximation

$$\frac{d\vec{\xi}^{\,\vec{\mu}}}{dr} = \mathbf{A}(\vec{\mu})\vec{\xi}^{\,\vec{\mu}}, \tag{28}$$

where $\vec{\mu}$ is a vector of parameters. For a given vector of parameters $\mu_j \in \vec{\mu}$, $j \in \{1, 2, \ldots, n\}$, HF solutions are again calculated at a few snapshots. Individual classical SLMs are subsequently performed upon these snapshots.

We would like to use our algorithm to predict at test sets of parameters; we denote these as $\vec{\theta}$. The B-GRIM method is employed to identify the nearest neighbor of the test parameter θ from the training set of snapshots. This enables a targeted interpolation of the eigenpairs associated with $\mathbf{A}$. Using this

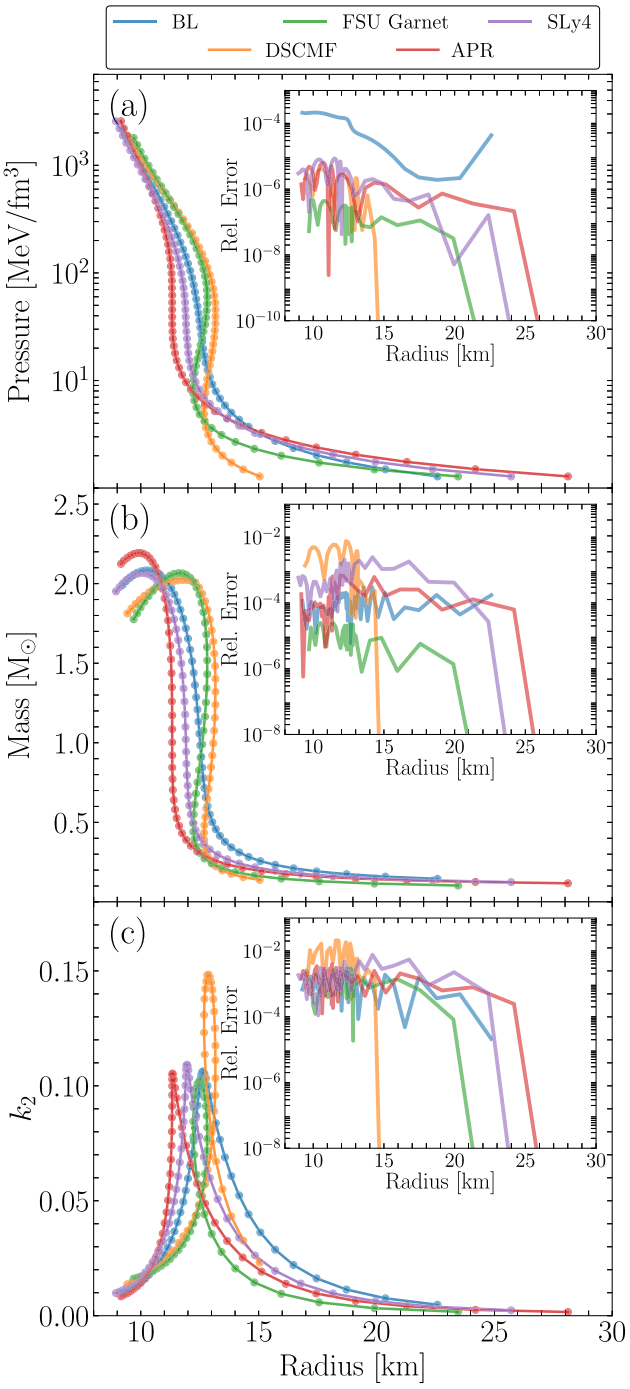


FIG. 2. (a) Central pressure, (b) mass, and (c) dimensionless Love number k_2 as functions of radius. These properties are calculated for five tabular EOSs [48,59–63]. The dots correspond to the HF results, and the solid lines indicate the SLM predictions. The inset shows the relative error between the HF and SLM results.

approach, we compute SVDs of J datasets by setting the accuracy using Eq. (17) and select the highest of the J ranks $[s = \min_{j\in J}(s_j)]$, yielding the reduced left- and right-singular matrices $\mathbf{U}_{js}$ and $\mathbf{V}_{js}^T$ along with the corresponding reduced singular value matrix $\mathbf{\Sigma}_{js}$. The individual reduced Koopman operators, $\mathbf{A}_{js}$, for each of the J cases are constructed as

$$\mathbf{A}_{js} = \mathbf{U}_{js}^T \mathbf{X}_j' \mathbf{V}_{js} \mathbf{\Sigma}_{js}^{-1}, \tag{29}$$

ALGORITHM 2. Parametric SLM. As with Algorithm 1, the italicized portions of this algorithm are direct extensions to Refs. [40,42].

1: Solve Eq. (26) for all training parameters $\vec{\mu}_j$ and collect N_s snapshots $\mathbf{X}_{\vec{\mu}_j} = [\vec{\xi}(r_i;\vec{\mu}_j)]_{i=0}^m$ for $1 \leqslant j \leqslant N_s$.
2: *Compute the logarithm of these snapshots* $[\log \vec{\xi}(r_i;\vec{\mu}_j)]_{i=0}^m$.
3: *Extend the snapshots by adding the Hadamard product of the arrays, i.e.,* $[\prod_{i,k} \log \vec{\xi}_i \log \vec{\xi}_k]_j$, $1 \leqslant j \leqslant J$.
4: Organize the snapshots in $\mathbf{X}_j^+$ and $\mathbf{X}_j^-$ data matrices, $1 \leqslant j \leqslant J$.
5: Perform SVD of $\mathbf{X}_j^-$: $\mathbf{X}_j^- = \mathbf{U}_j \mathbf{\Sigma}_j \mathbf{V}_j^T$, $1 \leqslant j \leqslant J$.
6: Retain s modes and compute the reduced Koopman operator $\mathbf{A}_{js} \equiv \mathbf{U}_{js}^T \mathbf{A} \mathbf{U}_{js} = \mathbf{U}_{js}^T \mathbf{X}_j^+ \mathbf{V}_{js} \mathbf{\Sigma}_{js}^{-1}$, $1 \leqslant j \leqslant J$.
7: Perform the eigendecomposition of $\mathbf{A}_{js}$ to obtain the reduced eigenmodes as described in Eq. (30).
8: Interpolate SVD modes $\mathbf{U}_j$, eigenmodes $\mathbf{W}_{js}$, and eigenvalues $\mathbf{\Lambda}_{js}$ to obtain $\mathbf{U}_{\theta s}$, $\mathbf{W}_{\theta s}$, and $\mathbf{\Lambda}_{\theta s}$, respectively, *using the B-GRIM interpolation method.*
9: Construct the modes $\mathbf{\Phi}_\theta = \mathbf{U}_{\theta s} \mathbf{W}_{\theta s}$.
10: Recover the initial coefficients $\vec{b}_\theta$ by interpolating $\vec{b}_j$, $1 \leqslant j \leqslant J$.
11: Reconstruct the coupled set of solutions *and exponentiate* to obtain $\vec{\xi}^{\,\theta}(r)$.

from which the reduced eigenpairs are obtained as

$$\mathbf{A}_{js} \mathbf{W}_{js} = \mathbf{\Lambda}_{js} \mathbf{W}_{js}. \tag{30}$$

The desired test case eigenpair $(\mathbf{\Lambda}_{\theta s}, \mathbf{W}_{\theta s})$ and $\mathbf{U}_{\theta s}$ and $\vec{b}_\theta$ are then determined via interpolation using B-GRIM, leveraging the J eigenpairs $(\mathbf{\Lambda}_{js}, \mathbf{W}_{js})$ [65]. The projected modes are subsequently calculated as $\mathbf{\Phi}_\theta = \mathbf{U}_{\theta s} \mathbf{W}_{\theta s}$. Finally, we obtain the solutions for the test parameter set $\vec{\theta}$ using Eq. (22) and keeping s modes. The corresponding full algorithm for pSLM is given in Algorithm 2.

B. Results

To test our ability to generate solutions to parametric EOSs at various parameter sets, we use the quarkyonic EOS [66] to test the pSLM algorithm. This EOS assumes that both nucleons and quarks are quasiparticles and that quarks reside in the Fermi sea, while at a sufficiently large Fermi energy, nucleons appear as correlations at the edge of the Fermi surface. In momentum space, this corresponds to a sphere of quarks surrounded by a shell of baryons of width Δ. To find this width, we use the expression

$$\Delta = \frac{\Lambda^3}{k_{FB}^2} + \kappa \frac{\Lambda}{N_c^2}, \tag{31}$$

where N_c is the number of quark colors and k_{FB} is the Fermi momentum of the baryons. The two unknown parameters above, Λ and κ, must be set to obtain the EOS. Hence, we vary these two parameters to produce various EOSs from this quarkyonic prescription.

We vary Λ and κ over a range of 300–500 MeV and 0.1–0.3, respectively. The training set for the pSLM routine is composed of 18 datasets, each composing 50 snapshots consisting of radii, central pressures, masses, and Love numbers. Using the pSLM algorithm, we predict the curves for test

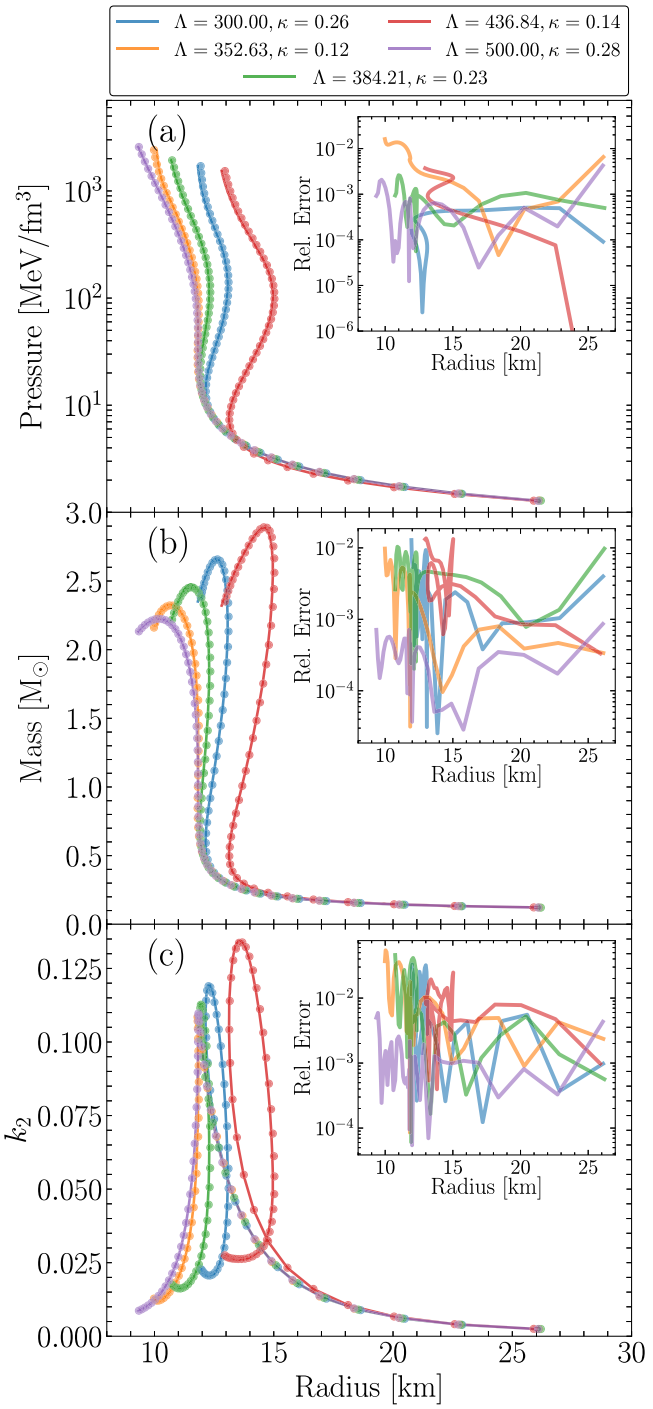


FIG. 3. (a) Central pressure, (b) mass, and (c) dimensionless Love number k_2 as functions of radius for the quarkyonic EOS [66] using pSLM, with varying parameters Λ and κ. The result shown here is the prediction for a given test parameter set that has not been used to train the emulator. The dots (solid lines) indicate SLM (HF) results.

parameter sets as shown in Fig. 3, which were not used to train the emulator.

The maximum mass and radius results from the HF solver and the pSLM emulator, as well as the percent error and speed of the calculation, are also recorded in Table I. The average speed for the pSLM emulator is $\approx 1.63 \times 10^{-4}$ s, yielding an overall average speedup of $\approx 7.0 \times 10^4$ over the HF RK4 solver. All percent errors are $<1\%$ for all maximum masses and radii.

VI. COMPARISON TO OTHER SOLVERS

To provide a comparison of the SLM and pSLM methods to other differential equation solvers, we choose two finite difference methods, namely, the forward Euler method and RK2 (Heun's) method. These were chosen because they use one and two function evaluations per iteration, respectively, compared to the four used by RK4, and hence constitute a "lower-fidelity" solution. We note that all three of these solvers employ a fixed step size. The TOV solutions using these methods are shown in Tables II–IV, where we compare them to our SLM and pSLM results for each EOS.

To improve the accuracy of these low-fidelity solvers, twice the number of points were chosen. This results in an effective doubling of the run-time for the three solvers, as seen in Tables II–IV. In Table IV, we also show results for the HF RK4 method using double the number of points as in the previous sections to make a fair comparison between the methods in this subsection. As expected, there is a speedup of about three times in using the RK2 and Euler methods compared to the standard RK4 method. However, looking at the errors of these two methods compared to the SLM errors with respect to the HF solutions, SLM performs much better than the RK2 and Euler methods. Additionally, using these other, lower-fidelity methods does not affect the order of magnitude of the SLM or pSLM run times.

VII. SUMMARY AND OUTLOOK

In this work, we presented SLM as a method to emulate the TOV equations, including tidal deformabilities. We employed SLM to emulate M–R curves from tabular EOSs, such as the well-known SLy4 EOS. We then used a parametric version of SLM (pSLM) to emulate EOSs that possess an arbitrary number of parameters. In particular, we focused on the quarkyonic EOS. The results from these emulators agree well with HF RK4 evaluations of the TOV equations. We gain an average computational speedup of $\sim 4.7 \times 10^4$ across both types of EOSs considered; hence, both SLM algorithms present highly efficient emulators for large-scale computations of the TOV equations in scenarios such as those presented by multimessenger astrophysical inference frameworks. This emulation strategy will help to reduce the bottleneck that the TOV equations present in such calculations, without sacrificing necessary accuracy.

Our SLM algorithm is not limited to small parameter sets and as such can be easily generalized to EOSs with larger parameter sets. More broadly, the SLM algorithm works well for problems that possess logarithmic dependence in the solutions. Hence, our work can be straightforwardly extended in future to problems that share this trait, e.g., density functionals with nonaffine parameter dependence, or time-dependent nuclear structure observables.

TABLE II. Similar to Table I, except the forward Euler method is used. The number of points has been doubled to ensure stable and accurate solutions.

	Euler		SLM		Rel. error (%)		Time (s)	
EOS	Max. mass ($M_\odot$)	Radius (km)	Max. mass ($M_\odot$)	Radius (km)	Max. mass	Radius	Euler	SLM
				Tabular EOS				
SLy4	2.067	9.979	2.065	9.977	0.06	0.02	5.748	4.830×10^{-4}
APR	2.192	9.932	2.193	9.931	0.01	0.01	5.542	4.807×10^{-4}
FSU Garnet	2.066	11.547	2.066	11.548	0.00	0.01	5.551	5.269×10^{-4}
BL	2.082	10.294	2.083	10.297	0.00	0.03	5.540	4.621×10^{-4}
DS-CMF-5	2.023	11.755	2.031	11.548	0.44	1.76	5.548	3.998×10^{-4}
				Quarkyonic EOS				
$\Lambda = 300.00, \kappa = 0.26$	2.887	14.482	2.645	12.936	0.77	2.55	5.559	1.898×10^{-4}
$\Lambda = 352.63, \kappa = 0.12$	2.666	12.566	2.321	10.585	0.33	0.81	5.521	1.662×10^{-4}
$\Lambda = 373.68, \kappa = 0.23$	2.476	11.467	2.470	11.639	0.25	1.06	5.489	1.469×10^{-4}
$\Lambda = 436.84, \kappa = 0.14$	2.329	10.616	2.902	14.520	0.52	0.08	5.599	1.631×10^{-4}
$\Lambda = 500.00, \kappa = 0.28$	2.223	10.12	2.281	10.434	2.59	2.53	5.473	2.360×10^{-4}

TABLE III. Similar to Table I, except the RK2 (Heun's) method is used. As in Table II, the number of points has been doubled to ensure stable and accurate solutions.

	RK2		SLM		Rel. error (%)		Time (s)	
EOS	Max. mass ($M_\odot$)	Radius (km)	Max. mass ($M_\odot$)	Radius (km)	Max. mass	Radius	RK2	SLM
				Tabular EOS				
SLy4	2.067	10.046	2.067	10.054	0.01	0.08	11.075	4.592×10^{-4}
APR	2.193	9.999	2.193	9.998	0.01	0.01	11.362	5.550×10^{-4}
FSU Garnet	2.066	11.607	2.067	11.607	0.03	0.00	11.105	4.551×10^{-4}
BL	2.083	10.361	2.083	10.361	0.01	0.00	11.145	5.572×10^{-4}
DS-CMF-5	2.023	11.815	2.031	11.611	0.41	1.73	11.542	3.970×10^{-4}
				Quarkyonic EOS				
$\Lambda = 300.00, \kappa = 0.26$	2.887	14.536	2.668	13.011	0.09	3.14	11.433	1.459×10^{-4}
$\Lambda = 352.63, \kappa = 0.12$	2.666	12.626	2.321	10.649	0.35	0.21	11.449	1.583×10^{-4}
$\Lambda = 373.68, \kappa = 0.23$	2.476	11.527	2.450	11.668	1.07	1.31	11.273	1.769×10^{-4}
$\Lambda = 436.84, \kappa = 0.14$	2.329	10.683	2.891	14.585	0.14	0.36	11.095	1.471×10^{-4}
$\Lambda = 500.00, \kappa = 0.28$	2.224	10.187	2.281	10.495	2.60	3.13	11.074	2.100×10^{-4}

TABLE IV. Similar to Table I, except the HF RK4 method uses twice the number of points than those in Table I for a direct comparison with Tables II and III.

	RK4		SLM		Rel. error (%)		Time (s)	
EOS	Max. mass ($M_\odot$)	Radius (km)	Max. mass ($M_\odot$)	Radius (km)	Max. mass	Radius	RK4	SLM
				Tabular EOS				
SLy4	2.067	10.033	2.066	10.034	0.03	0.02	22.519	4.580×10^{-4}
APR	2.193	9.986	2.195	9.987	0.10	0.01	22.515	4.930×10^{-4}
FSU Garnet	2.066	11.601	2.066	11.601	0.00	0.00	22.629	4.508×10^{-4}
BL	2.083	10.348	2.083	10.352	0.01	0.05	22.536	4.742×10^{-4}
DS-CMF-5	2.023	11.808	2.034	11.599	0.57	1.78	22.507	4.930×10^{-4}
				Quarkyonic EOS				
$\Lambda = 300.00, \kappa = 0.26$	2.887	14.529	2.660	12.993	0.21	3.00	23.135	1.621×10^{-4}
$\Lambda = 352.63, \kappa = 0.12$	2.666	12.612	2.320	10.643	0.39	0.27	22.808	1.569×10^{-4}
$\Lambda = 373.68, \kappa = 0.23$	2.476	11.520	2.452	11.661	0.98	1.25	22.607	1.438×10^{-4}
$\Lambda = 436.84, \kappa = 0.14$	2.329	10.669	2.892	14.582	0.17	0.34	22.751	1.359×10^{-4}
$\Lambda = 500.00, \kappa = 0.28$	2.224	10.173	2.286	10.488	2.79	3.06	22.797	2.222×10^{-4}

ACKNOWLEDGMENTS

We thank Jérôme Margueron for a careful review of our manuscript, Xilin Zhang for engaging discussions, and Amy L. Anderson for providing the FSU Garnet data. We also thank `CompOSE` [47] for providing all other tabular EOS datasets. A.C.S. thanks the Facility for Rare Isotope Beams for their hospitality and encouragement during the completion of this work. This work was supported by the CSSI program, Award No. OAC-2004601 (BAND collaboration [67]) (A.C.S.), and the U.S. Department of Energy, Office of Science, Nuclear Physics, under Awards No. DE-SC0023688 (S.L.), No. DE-FG02-93-40756 (A.C.S.), and No. DE-SC0024233 (STREAMLINE collaboration) (J.M.M.).

DATA AVAILABILITY

To help other researchers utilize and build upon our work, the code that contains the SLM algorithm and produced the results in this paper, as well as tutorials to assist new users, is published in a public GitHub repository for the use of the scientific community [44].

APPENDIX A: SCALING TOV AND TIDAL EQUATIONS

The TOV equations [17,18], along with the tidal deformability equation [20–22], are given by [68]

$$\frac{dP}{dr} = -\frac{G}{c^2}[\epsilon(r)+P(r)]\frac{m(r)+4\pi r^3 P(r)/c^2}{r[r-2Gm(r)/c^2]}, \tag{A1}$$

$$\frac{dm}{dr} = 4\pi r^2 \frac{\epsilon(r)}{c^2}, \tag{A2}$$

$$\frac{dy}{dr} = -\frac{y(r)^2}{r} - \frac{F(r)y(r)}{r} - \frac{Q(r)}{r}, \tag{A3}$$

where

$$F(r) = \frac{1-4\pi G r^2[\epsilon(r)-P(r)]/c^4}{1-2\frac{Gm(r)}{rc^2}} \tag{A4}$$

and

$$Q(r) = \frac{4\pi G r^2/c^4}{1-\frac{2Gm(r)}{rc^2}} \times \left(5\epsilon(r)+9p(r)+\frac{\epsilon(r)+p(r)}{c_s(r)^2}c^2 - \frac{6c^4}{4\pi r^2 G}\right) - 4\left(\frac{G[m(r)/(rc^2)+4\pi r^2 p(r)/c^4]}{1-2Gm(r)/(rc^2)}\right)^2. \tag{A5}$$

Following closely Ref. [46], we use the scaling $r = R_0 x$, $m = M_0 m(x)$, $P = P_0 p(x)$, and $\epsilon = \varepsilon_0 \varepsilon(x)$. To define the relevant scales, we set

$$\varepsilon_0 = P_0 \equiv \frac{1}{8\pi^2}\frac{(m_n c^2)^4}{(\hbar c)^3} \approx 1.285\,\mathrm{GeV/fm^3} \tag{A6}$$

and adopt the natural normalization

$$\left[\frac{2GM_0}{c^2 R_0}\right] = \left[\frac{4\pi R_0^3 \varepsilon_0}{3M_0 c^2}\right] = 1. \tag{A7}$$

Setting these quantities to unity, meaning to a dimensionless value of order one, establishes "natural" length and mass scales for the problem. This leads to

$$R_0 = \sqrt{\frac{3\pi}{\alpha_G}}\lambda_n \approx 8.378\,\mathrm{km}, \tag{A8}$$

$$M_0 = \left(\frac{R_0}{R_s^\odot}\right)M_\odot \approx 2.837\,M_\odot, \tag{A9}$$

where α_G is the small, dimensionless gravitational coupling strength between two neutrons (or the neutron mass to Planck mass ratio), λ_n is the Compton wavelength of the neutron, and $R_s^\odot$ represents the Schwarzschild radius of the Sun. Numerically, these values are

$$\alpha_G = \frac{Gm_n^2}{\hbar c} \approx 5.922\times 10^{-39}, \tag{A10}$$

$$\lambda_n = \frac{\hbar c}{m_n c^2} \approx 0.210\times 10^{-18}\,\mathrm{km}, \tag{A11}$$

$$R_s^\odot = \frac{2GM_\odot}{c^2} \approx 2.953\,\mathrm{km}. \tag{A12}$$

The resulting scaled equations are then given by

$$\frac{dp}{dx} = -\frac{1}{2}\frac{[\varepsilon(x)+p(x)][m(x)+3x^3 p(x)]}{x^2[1-m(x)/x]}, \tag{A13}$$

$$\frac{dm}{dx} = 3x^2\varepsilon(x), \tag{A14}$$

$$\frac{dy}{dx} = -\frac{y(x)^2}{x} - \frac{F(x)y(x)}{x} - \frac{Q(x)}{x}, \tag{A15}$$

with

$$F(x) = \frac{1-\frac{3}{2}x^2[\varepsilon(x)-p(x)]}{1-\frac{m(x)}{x}}$$

and

$$Q(x) = \frac{\frac{3}{2}x^2}{\left(1-\frac{m(x)}{x}\right)}\left[5\varepsilon(x)+9p(x)+\frac{\varepsilon(x)+p(x)}{c_s^2(x)}-\frac{4}{x^2}\right] - \left(\frac{m(x)/x+3x^2 p(x)}{\left(1-\frac{m(x)}{x}\right)}\right)^2.$$

Solving these equations, we get the scaled $m(x)$, $p(x)$, $y(x)$. To calculate the mass-radius curve of a neutron star, one enforces the boundary condition that the pressure of star is zero at the surface. This enables one to find the mass and radius of the star for a given central pressure.

We use $y_R = y(r=R)$, where R is the radius of the star with mass $M[M_\odot]$ to calculate the compactness of the star $\beta = GM/R$. Using this, one can then calculate the dimensionless Love number k_2 [see Eq. (11)].

APPENDIX B: BANACH GREEDY RECOMBINATION INTERPOLATION METHOD (B-GRIM)

In this approach [64], the goal is to find sparse approximations of functions thereby reducing computational complexity. The objective is to construct an interpolant that minimizes the norm (ℓ^1) in that space. In the greedy recombination method, the algorithm iteratively selects basis elements to improve the

interpolation, minimizing the error at each step. The method is composed of two main steps: (1) Banach extension and (2) Banach recombination.

1. Banach extension

Let $u \in \mathrm{Span}(\mathcal{F})$, where $\mathcal{F}$ is the set of functionals in the training set corresponding to features that approximate function $\boldsymbol{y}$, and assume $L \subset \Sigma$ be a finite collection of linear functionals, where Σ denotes the data. For each subset $L \subset \Sigma$, the method employs a recombination process to find an approximation $u \in \mathrm{Span}(\mathcal{F})$ for the target function $\boldsymbol{y}$, where $\boldsymbol{y}$ represents the desired solution constrained by L. Typically, this process introduces numerical errors; to minimize these, it is assumed that u is sufficiently close to $\boldsymbol{y}$ at each functional $\sigma \in L$, a relationship that is formalized in the recombination step.

2. Banach recombination

Banach recombination phase consists of three steps.

(1) Choose a subset $L_j \subset \Sigma$ by randomly permuting the element order in L.

(2) Perform recombination thinning to find an element $u_j \in \mathrm{Span}(\mathcal{F})$ that satisfies $|\sigma(\boldsymbol{y} - u_j)| \leqslant \varepsilon_0$ for every $\sigma \in L_j$, where ε_0 is a small tolerance value.

(3) Compute the error metric $E[u_j] := \max\{|\sigma(\boldsymbol{y} - u_j)| : \sigma \in \Sigma\}$.

After identifying the elements $u_1, \ldots, u_s$, define u as

$$u := \mathrm{argmin}\, E[w] : w \in \{u_1, \ldots, u_s\}. \tag{B1}$$

This final u serves as the best approximation of the target function $\boldsymbol{y}$. The full algorithm and derivation of the method is provided in Ref. [64].

[1] B. P. Abbott *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), GW170817: Observation of gravitational waves from a binary neutron star inspiral, Phys. Rev. Lett. **119**, 161101 (2017).

[2] B. P. Abbott *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), GW170817: Measurements of neutron star radii and equation of state, Phys. Rev. Lett. **121**, 161101 (2018).

[3] B. P. Abbott *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), GW190425: Observation of a compact binary coalescence with total mass $\sim$3.4 $M_\odot$, Astrophys. J. Lett. **892**, L3 (2020).

[4] 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**, L24 (2019).

[5] M. C. Miller *et al.*, The radius of PSR J0740+6620 from NICER and XMM-Newton data, Astrophys. J. Lett. **918**, L28 (2021).

[6] T. E. Riley *et al.*, A NICER view of PSR J0030+0451: Millisecond pulsar parameter estimation, Astrophys. J. Lett. **887**, L21 (2019).

[7] T. E. Riley *et al.*, A NICER view of the massive pulsar PSR J0740+6620 informed by radio timing and XMM-Newton spectroscopy, Astrophys. J. Lett. **918**, L27 (2021).

[8] D. Choudhury *et al.*, A NICER view of the nearest and brightest millisecond pulsar: PSR J0437–4715, Astrophys. J. Lett. **971**, L20 (2024).

[9] D. R. Phillips, R. J. Furnstahl, U. Heinz, T. Maiti, W. Nazarewicz, F. M. Nunes, M. Plumlee, M. T. Pratola, S. Pratt, F. G. Viens, and S. M. Wild, Get on the BAND wagon: A Bayesian framework for quantifying model uncertainties in nuclear dynamics, J. Phys. G **48**, 072001 (2021).

[10] J. A. Melendez, R. J. Furnstahl, D. R. Phillips, M. T. Pratola, and S. Wesolowski, Quantifying correlated truncation errors in effective field theory, Phys. Rev. C **100**, 044001 (2019).

[11] B. A. Brown *et al.*, Motivations for early high-profile FRIB experiments, J. Phys. G **52**, 050501 (2025).

[12] A. Sorensen *et al.*, Dense nuclear matter equation of state from heavy-ion collisions, Prog. Part. Nucl. Phys. **134**, 104080 (2024).

[13] R. Kumar *et al.* (MUSES Collaboration), Theoretical and experimental constraints for the equation of state of dense and hot matter, Living Rev. Relativ. **27**, 3 (2024).

[14] C. Drischler, J. W. Holt, and C. Wellenhofer, Chiral effective field theory and the high-density nuclear equation of state, Annu. Rev. Nucl. Part. Sci. **71**, 403 (2021).

[15] J. M. Lattimer, Neutron stars and the nuclear matter equation of state, Annu. Rev. Nucl. Part. Sci. **71**, 433 (2021).

[16] H. Koehn *et al.*, From existing and new nuclear and astrophysical constraints to stringent limits on the equation of state of neutron-rich dense matter, Phys. Rev. X **15**, 021014 (2025).

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

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

[19] K. S. Thorne and A. Campolattaro, Non-radial pulsation of general-relativistic stellar models. I. Analytic analysis for $l \geqslant 2$, Astrophys. J. **149**, 591 (1967).

[20] T. Hinderer, Tidal Love numbers of neutron stars, Astrophys. J. **677**, 1216 (2008); **697**, 964(E) (2009).

[21] 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**, 123016 (2010).

[22] S. Postnikov, M. Prakash, and J. M. Lattimer, Tidal Love numbers of neutron and self-bound quark stars, Phys. Rev. D **82**, 024016 (2010).

[23] Ch. C. Moustakidis, T. Gaitanos, Ch. Margaritis, and G. A. Lalazissis, Bounds on the speed of sound, tidal polarizability and gravitational waves from neutron stars, Bulg. J. Phys. **44**, S93 (2017).

[24] P. T. H. Pang *et al.*, An updated nuclear-physics and multi-messenger astrophysics framework for binary neutron star mergers, Nat. Commun. **14**, 8352 (2023).

[25] R. Abbott *et al.* (LIGO Scientific Collaboration, Virgo Collaboration, and KAGRA Collaboration), Population of merging compact binaries inferred using gravitational waves through GWTC-3, Phys. Rev. X **13**, 011048 (2023).

[26] C. Drischler, J. A. Melendez, R. J. Furnstahl, and D. R. Phillips, Quantifying uncertainties and correlations in the nuclear-matter equation of state, Phys. Rev. C **102**, 054315 (2020).

[27] 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**, 202702 (2020).

[28] C. Drischler, S. Han, J. M. Lattimer, M. Prakash, S. Reddy, and T. Zhao, Limiting masses and radii of neutron stars and their implications, Phys. Rev. C **103**, 045808 (2021).

[29] A. C. Semposki, C. Drischler, R. J. Furnstahl, J. A. Melendez, and D. R. Phillips, From chiral effective field theory to perturbative QCD: A Bayesian model mixing approach to symmetric nuclear matter, Phys. Rev. C **111**, 035804 (2025).

[30] B. Hu *et al.*, *Ab initio* predictions link the neutron skin of ^{208}Pb to nuclear forces, Nat. Phys. **18**, 1196 (2022).

[31] B. T. Reed, R. Somasundaram, S. De, C. L. Armstrong, P. Giuliani, C. Capano, D. A. Brown, and I. Tews, Towards accelerated nuclear-physics parameter estimation from binary neutron star mergers: Emulators for the Tolman-Oppenheimer-Volkoff equations, Astrophys. J. **974**, 285 (2024).

[32] P. T. H. Pang, L. Sivertsen, R. Somasundaram, T. Dietrich, S. Sen, I. Tews, M. W. Coughlin, and C. Van Den Broeck, Probing quarkyonic matter in neutron stars with the Bayesian nuclear-physics multimessenger astrophysics framework, Phys. Rev. C **109**, 025807 (2024).

[33] C. Drischler, J. A. Melendez, R. J. Furnstahl, A. J. Garcia, and X. Zhang, BUQEYE guide to projection-based emulators in nuclear physics, Front. Phys. **10**, 1092931 (2022).

[34] E. Bonilla, P. Giuliani, K. Godbey, and D. Lee, Training and projecting: A reduced basis method emulator for many-body physics, Phys. Rev. C **106**, 054322 (2022).

[35] J. Maldonado, R. J. Drischler, C. Furnstahl, and P. Mlinarić, Greedy emulators for nuclear two-body scattering, Phys. Rev. C **112**, 024002 (2025).

[36] P. Giuliani, K. Godbey, E. Bonilla, F. Viens, and J. Piekarewicz, Bayes goes fast: Uncertainty quantification for a covariant energy density functional emulated by the reduced basis method, Front. Phys. **10**, 1054524 (2023).

[37] D. Odell, P. Giuliani, K. Beyer, M. Catacora-Rios, M. Y. H. Chan, E. Bonilla, R. J. Furnstahl, K. Godbey, and F. M. Nunes, ROSE: A reduced-order scattering emulator for optical models, Phys. Rev. C **109**, 044612 (2024).

[38] 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**, 055803 (2020).

[39] J. H. Tu, C. K. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, On dynamic mode decomposition: Theory and applications, J. Comput. Dyn. **1**, 391 (2014).

[40] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016).

[41] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, Modern Koopman theory for dynamical systems, arXiv:2102.12086.

[42] Q. A. Huhn, M. E. Tano, J. C. Ragusa, and Y. Choi, Parametric dynamic mode decomposition for reduced order modeling, J. Comput. Phys. **475**, 111852 (2023).

[43] N. Demo, M. Tezzele, and G. Rozza, Pydmd: Python dynamic mode decomposition, J. Open Source Softw. **3**, 530 (2018).

[44] S. Lalit, A. C. Semposki, and J. M. Maldonado, Star Log-extended eMulator repository, GitHub, 2024, https://github.com/asemposki/SLM.

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

[46] J. Piekarewicz, Neutron star matter equation of state, in *Handbook of Supernovae*, edited by Athem W. Alsabti and Paul Murdin (Springer International Publishing, Cham, 2017), pp. 1075–1094.

[47] S. Typel *et al.* (CompOSE Core Team), CompOSE reference manual, Eur. Phys. J. A **58**, 221 (2022).

[48] W.-C. Chen and J. Piekarewicz, Searching for isovector signatures in the neutron-rich oxygen and calcium isotopes, Phys. Lett. B **748**, 284 (2015).

[49] B. W. Brunton, L. A. Johnson, J. G. Ojemann, and J. N. Kutz, Extracting spatial–temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition, J. Neurosci. Methods **258**, 1 (2016).

[50] J. Mann and J. Nathan Kutz, Dynamic mode decomposition for financial trading strategies, Quant. Financ. **16**, 1643 (2016).

[51] K. Jiaqing and Z. Weiwei, Dynamic mode decomposition and its applications in fluid dynamics, ACTA Aerodyn. Sin. **36**, 163 (2018).

[52] J. A. Melendez, C. Drischler, R. J. Furnstahl, A. J. Garcia, and X. Zhang, Model reduction methods for nuclear emulators, J. Phys. G **49**, 102001 (2022).

[53] M. Korda and I. Mezić, On convergence of extended dynamic mode decomposition to the Koopman operator, J. Nonlinear Sci. **28**, 687 (2018).

[54] P. J. Baddoo, B. Herrmann, B. J. McKeon, J. N. Kutz, and S. L. Brunton, Physics-informed dynamic mode decomposition, Proc. R. Soc. A: Math. Phys. Eng. Sci. **479**, 20220576 (2023).

[55] M. Velegar, C. Keller, and J. N. Kutz, Optimized dynamic mode decomposition for reconstruction and forecasting of atmospheric chemistry data, arXiv:2404.12396.

[56] M. N. Sharak, A. Safavinejad, and M. K. Moayyedi, A fast method based on dynamic mode decomposition for radiative heat transfer in participating media, J. Quant. Spectrosc. Radiat. Transfer **288**, 108248 (2022).

[57] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition, J. Nonlinear Sci. **25**, 1307 (2015).

[58] To construct a mass-radius curve from these SLM solutions, we can impose the condition $p = 0$ at $r = R$ and collect the values of the mass [$M = m(r = R)$] [and $y(r = R)$ for the tidal deformability, if desired]. Hence, SLM can be used to emulate the profiles of the EOS and the result can be propagated to the final M–R curve.

[59] I. Bombaci and D. Logoteta, Equation of state of dense nuclear matter and neutron star structure from nuclear chiral interactions, Astron. Astrophys. **609**, A128 (2018).

[60] E. Chabanat, P. Bonche, P. Haensel, J. Meyer, and R. Schaeffer, A Skyrme parametrization from subnuclear to neutron star densities, Nucl. Phys. A **627**, 710 (1997).

[61] V. Dexheimer, R. O. Gomes, T. Klähn, S. Han, and M. Salinas, GW190814 as a massive rapidly rotating neutron star with exotic degrees of freedom, Phys. Rev. C **103**, 025808 (2021).

[62] V. Dexheimer and S. Schramm, Proto-neutron and neutron stars in a chiral SU(3) model, Astrophys. J. **683**, 943 (2008).

[63] A. Akmal, V. R. Pandharipande, and D. G. Ravenhall, Equation of state of nucleon matter and neutron star structure, Phys. Rev. C **58**, 1804 (1998).

[64] T. Lyons and A. D. McLeod, Greedy recombination interpolation method (GRIM), arXiv:2205.07495.

[65] B-GRIM is optimal for high-dimensional spaces where residual minimization in Banach spaces is effective. This is especially true when dealing with noisy data or requiring iterative improvement of the solution, whereas Lagrange interpolation is unstable in such cases (see Appendix B for more details).

[66] L. McLerran and S. Reddy, Quarkyonic matter and neutron stars, Phys. Rev. Lett. **122**, 122701 (2019).

[67] K. Beyer *et al.*, BANDFramework: An open-source framework for Bayesian analysis of nuclear dynamics, Version 0.4.0, 2024, https://github.com/bandframework/bandframework.

[68] For a detailed derivation of these equations, see Refs. [20–22,69].

[69] A. F. Fantina, J. L. Zdunik, N. Chamel, J. M. Pearson, L. Suleiman, and S. Goriely, Accreting neutron stars from the nuclear energy-density functional theory—II. Equation of state and global properties, Astron. Astrophys. **665**, A74 (2022).