Efficient Learning of Accurate Surrogates for Simulations of Complex Systems

A. Diaw[1,2,*], M. McKerns[1], I. Sagert[1], L. G. Stanton[3], and M. S. Murillo[4]

[1]Los Alamos National Laboratory, Los Alamos, NM USA 87545
[2]RadiaSoft LLC, Boulder, CO 80301 USA
[3]Department of Mathematics and Statistics, San José State University, San José, CA 95192 USA
[4]Department of Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, MI 48824

ABSTRACT

Machine learning and interpolation methods are commonly used to build computationally inexpensive surrogates for physical models. However, the predictive capability of these approaches can suffer when data are sparse, or the model is nonlinear. To overcome this problem, we introduce an online learning method empowered by optimizer-driven sampling. This method presents two advantages to current approaches. First, it ensures that the critical points of the model are included in the training data. Second, the surrogates are tested and retrained after any new model evaluation when a defined validity measure drops below a specified threshold. We assess the performance of our method on benchmark functions and find that optimizer-directed sampling generally outperforms in terms of accuracy in critical regions, which holds even when the scoring metric favors overall accuracy. Then, we apply our method to nuclear matter, demonstrating that highly accurate surrogates for the nuclear equation of state can be reliably auto-generated as calculated by expensive simulations using a minimum of model evaluations.

Introduction

Science and engineering applications are increasingly dependent on inexpensive surrogates to represent complex physical systems. The surrogate models should come with some guarantee that they can reliably predict the system's behavior. In addition, there is a growing strategic need for tools that can robustly forecast the behavior of physical systems where data is high-dimensional, noisy, or sparse, the system model time-dependent and/or include uncertainty. For example in materials science[1], macroscopic simulations rely on closure information that is based on data from microphysical methods[2] such as Molecular Dynamics[3,4] or Monte Carlo[5,6] calculations. A fundamental challenge is to reach a level of predictive ability that is similar to the ones of the high-fidelity microscopic methods and can include complex phenomena like phase transitions, material mixtures, or shocks. In general, a significant amount of data needs to be generated from expensive microscopic models to enable statistically valid descriptions of complex macroscopic systems. Producing sufficient data however can require a prohibitively large number of microscopic calculations. This leads to potential roadblocks for materials discovery and design, and the robust prediction of material properties[7,8].

Such bottlenecks are not unique to materials science but are especially prevalent whenever robust predictions are relying on multi-scale phenomena. Climate modeling, quantum information science, and the automated control of instrumentation all require alternatives to expensive simulations for robust predictions[9–12]. As a consequence, researchers have begun to turn to the learning of surrogates for complex systems. The generation of such models through interpolation or machine learning holds great promise to overcome the often unfeasible brute-scale computational approaches and help enable the discovery of new science.

In recent studies, Lubbers et al.[13] and Diaw et al.[14] apply active learning to generate surrogates of fine-scale material response, while Roehm et al.[15] use kriging to construct surrogates of stress fields and communicate the results to a fine-scale code that solves the macro-scale conservation laws for elastodynamics. Noack et al.[12] use a similar kriging-based approach to construct surrogates for autonomous X-ray scattering experiments. None of the above studies ensures that the learned surrogates are valid on future data and thus cannot provide a guarantee for their validity. Instead, the surrogate evaluation is accompanied by the calculation of an uncertainty metric which determines if and where new fine-scale simulations should be launched. Noack et al.[12], for example, use a genetic algorithm to find the maximum of the variances for each measured data point and then draw new samples from a distribution that is localized around the solved maximum. Such passive approaches to validity assume that the fine-scale descriptions will always be available and suffer from frequent requests for the computationally expensive model evaluations.

Here we present an online learning methodology to efficiently construct surrogates that are *asymptotically* valid with respect to any future data. We choose this terminology as, while we do not have a formal proof, there is strong evidence of at least approximate validity for future data under some light conditions. More specifically, we conjecture that the minimum data set necessary to produce a highly accurate surrogate is composed of evaluations at all critical points on the model's response surface. Our claim comes with the condition that the selected class of surrogates has enough flexibility to reproduce the model accurately. Hence, we use a

radial basis function (RBF)[16] as the estimator when training a surrogate for the model's response surface. The utilization of RBFs arises from their universal capabilities for function approximation[17] and their connection to single hidden-layer feed-forward neural networks (NN) with non-sigmoidal nonlinearities[18]. Although a multilayer perceptron (MLP)[19] or another similar NN estimator are also potential choices, we will use RBF interpolation as it is generally more efficient for online learning[18]. Our methodology has three key components: (1) a sampling strategy to generate new training and test data, (2) a learning strategy to generate candidate surrogates from the training data, and (3) a validation metric to evaluate candidate surrogates against the test data. The numerical realization is done with *mystic*, an open-source optimization, learning, and uncertainty quantification toolkit[20,21]. For over a decade, *mystic* has been used for the optimization of complex models, including the usage of uncertainty metrics to optimally improve model accuracy and robustness[22–26] and increase the statistical robustness of surrogates[27–29]. Furthermore, recent developments include highly-configurable sampling strategies[30]. We will use these in combination with online learning to train surrogates for accuracy with respect to all future data.

The current work primarily focuses on how the choice of sampling strategy affects the efficiency when producing an asymptotically valid surrogate. Here, we characterize validity via the evolution of the surrogate test score. Fig. 1 shows the procedure to create such a surrogate for an expensive model. It is iterative and includes explicit validation and update mechanisms. To reduce computational complexity, we first link the model to a database (DB); thus, the input and output of the model are automatically stored when it is evaluated. Later, the DB of model evaluations is used to train candidate surrogates. When the model is evaluated, the corresponding surrogate is retrieved from the surrogate DB and tested for validity. If no stored surrogate exists, we skip testing and proceed directly to learning a candidate surrogate.

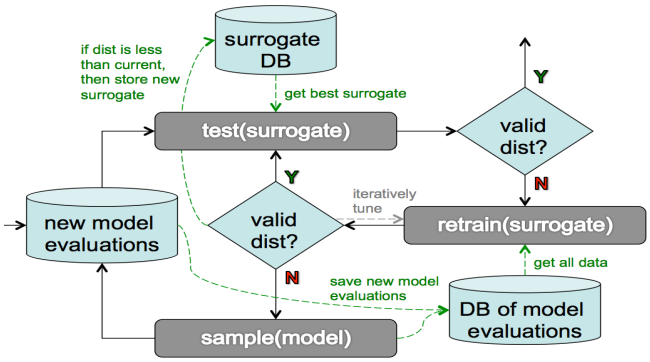


Figure 1. Automated generation of computationally inexpensive surrogates for complex physical systems. When new model evaluations occur, the corresponding surrogate is retrieved and evaluated for the same data. If the surrogate is determined to still be valid , execution stops. Otherwise, the surrogate is updated by retraining against the DB of stored model evaluations, where the surrogate is validated with a fine-tuning of surrogate hyperparameters against a quality metric. If iterative retraining improves the surrogate, it is saved. Otherwise, we sample new model evaluations to generate new data. The process repeats until testing produces a valid surrogate.

Results

Here we will assess the performance of different sampling strategies against benchmark functions. We then apply our methodology to finding accurate surrogates for equation-of-state (EOS) calculations of dense nuclear matter.

Numerical evaluation: benchmark functions

Our case studies use several benchmark functions that are commonly applied to test the performance of numerical optimization algorithms. We first examine how the sampling strategy affects the efficiency and effectiveness of finding an asymptotically valid surrogate. Then, we explore how the optimizer configuration impacts the efficiency of generating an initial valid surrogate.

Sampling for Asymptotic Validity. Here, we compare optimizer-directed sampling with random sampling regarding their ability to find an accurate surrogate for all future data. For this, we use the workflow for asymptotic validity to learn a surrogate for the d-dimensional Rastrigin function[31]:

$$f(\mathbf{x}) = 10\,d + \sum_{i=1}^{d}\left[x_i^2 - 10\,\cos(2\pi x_i)\right],\ x_i \in [0,10] \quad (1)$$

with $d = 2$. It is essentially a spherical function with an added cosine modulation that produces regularly distributed local minima. Our optimizer-directed sampler uses a "sparsity" sampling strategy with an ensemble of 16 `NelderMeadSimplexSolver` instances. We define our *test* for validity in Eq. (9) as:

$$\text{ave}(\Delta_y) \le tol_{ave} \ \wedge\ \max(\Delta_y) \le tol_{max}, \quad (2)$$

where $tol_{ave} = 10^{-5}$, $tol_{max} = 10^{-4}$, $\Delta_x \neq 0$ is a graphical distance, and *data* corresponds to all existing model evaluations (i.e. prior plus newly sampled). For *train* we also use Eq. (2) with $tol_{ave} = 10^{-5}$ and $tol_{max} = 10^{-4}$. A quality *metric* for training is given by $\delta = \sum_y \Delta_y$ and we define *converged* (see Eq. (12)) as:

$$\Omega(M) \ \vee\ \max_y\left(\max_j\left(\text{ave}(\Delta_{y,j})\right)\right) \le tol_{stop}\ , \quad (3)$$

with $\Omega(M)$ equal to "true" when no new local extrema have been found in the last $M = 3$ iterations. We use $tol_{stop} = 2 \cdot 10^{-4}$, $\Delta_{y,i}$ is the graphical distance to the *data* sampled in iteration i (i.e. no prior model evaluations), and j is given by the last $N = 3$ iterations $j \in [i - N + 1, ..., i]$. By using $warm = 1000$, we ensure that at least 1000 model evaluations

have been performed per iteration. In addition, we track the testing score for a single iteration i:

$$score = \text{ave}_y(\text{ave}(\Delta_{y,i})) \tag{4}$$

but do not use it to to terminate the calculation. To assess the effect of stricter tolerances, we repeat the calculation with $tol_{ave} = 10^{-7}$, $tol_{max} = 10^{-6}$, and $tol_{stop} = 2 \cdot 10^{-6}$. We will refer to these tolerance settings as "strict" and the prior tolerance settings as "loose". For both settings, we compare our results with pure systematic random sampling, using an ensemble of 500 points, after the initial and tenth iteration. The random sampling uses the default optimizer configuration and a metric based on the average surrogate misfit

For the 2D Rastrigin function, we find from Figures 2, 4b, and Table 1a that the test score for pure systematic random sampling converges faster than the optimizer-directed one. In addition, it yields an excellent representation of truth for both, strict and loose tolerances. Why is the random sampling strategy so successful in this case? The answer can be found in the characteristics of the Rastrigin function having shallow local extrema distributed uniformly across the response surface. A systematic random sampling approach which efficiently covers the input space can be expected to be more performant here than a strategy which attempts to pinpoint the extrema. However, physical setups rarely have a response surface with such a regular and frequent distribution of extrema.

With that, we perform a similar comparison for the Rosenbrock function[32]. Its 2D version is a saddle with an inverted basin which contains a long, narrow, and flat parabolic valley with global minima. The 8D Rosenbrock function is the sum of seven coupled 2D functions, with a global minimum at $x_i = 1$ and a local minima near $x = [-1, 1, ..., 1]$. In general, the saddle in the 2D Rosenbrock function is captured well with either systematic random or optimizer-directed sampling. Random sampling again converges quicker for strict and loose tolerances (see Table 1a). However, upon closer inspection in Fig. 4a we find that optimizer-directed sampling reproduces truth much more accurately in the region surrounding the global minimum. This is a direct consequence of the optimizer-directed strategy which provides a higher sampling density in the neighborhood of the minimum. We find this behavior for strict and loose tolerances and reproduce it for the 8D Rosenbrock function with loose tolerances (see Figures 4b and 3).

Finally, we use our approach for the 2D Michalewicz's function. It is generally flat but has several long narrow channels that have sharp dips at their intersections. We find that after $30,000$ model evaluations and when applying loose tolerances both, systematic random and optimizer-driven sampling, visually reproduce truth approximately. We expect that for stricter tolerances much more sampling will be required in order to generate a surrogate which reproduces the critical points with the same quality as for the previous benchmark functions. With that, the 2D Michalewicz function contains features that are challenging for both sampling approaches.

For all applied test functions, systematic random sampling is found to converge faster to an asymptotically valid surrogate. However, optimizer-directed sampling is superior in reproducing the behavior of a function at its extrema. In all cases, our optimizer was a Nelder-Mead in the default configuration. We expect that less strict convergence requirements will reduce the number of required evaluations, potentially at the cost of some accuracy in the vicinity of the extrema. We will explore the impact of optimizer configuration in the next section.

Sampling for Training Validity. Here, we assess the impact of the optimizer configuration on the efficiency of optimizer-directed sampling. Our sampler uses a lattice sampling strategy with an ensemble of 40 `NelderMeadSimplexSolver` instances. We define our *test* for validity as:

$$\sum_y \Delta_y \leq tol_{sum} \ \wedge \ \max(\Delta_y) \leq tol_{max}, \tag{5}$$

where $tol_{sum} = 10^{-3}$ and $tol_{max} = 10^{-6}$. We use a graphical distance with $\Delta_x \neq 0$, and *data* defined as all existing model evaluations (i.e. prior plus newly sampled). We define *train* as in Eq. (5), again with $tol_{sum} = 10^{-3}$ and $tol_{max} = 10^{-6}$. Finally, we use a quality *metric* for training, given by $\delta = \sum_y \Delta_y$, and define *converged*, in Eq. (12), identically to *test* in Eq. (5).

Table 1b gives the number of evaluations that is needed to obtain valid surrogates of several standard benchmark functions. We find that for the default optimizer configurations an ensemble of Nelder-Mead optimizers is more efficient, regardless of the validity of the benchmark function. This originates most likely from the Nelder-Mead solvers getting stuck in local extrema faster than those using Powell's method. Since the goal of the solver ensemble is to find as many local extrema as possible while using a minimum number of function evaluations, Nelder-Mead appears to be the better choice.

The following section will test the ability to quickly produce a valid surrogate that reproduces relevant physical behavior in regions where traditional methods have difficulty producing similar results. We will use a larger ensemble of optimizers and test the accuracy at the end of a single iteration of our entire workflow. This does not guarantee the surrogate will be valid against all future data but will give us an idea of how quickly the surrogates can accurately reproduce physical effects near the critical points.

Equation of State with Phase Transition

We are interested in building an accurate surrogate for an equation of state (EOS) for a dense nuclear matter that contains a phase transition (PT). Reliable hadronic models for nuclear matter exist up to baryon number densities n_b of about twice nuclear saturation density $n_0 \sim 0.16\,\text{fm}^{-3}$ and at asymptotically high densities of $n_b \gg 40\,n_0$[33,34]. While at low densities and temperatures T, the nuclear matter is composed of neutrons and protons, for high values of n_b and T, it is expected

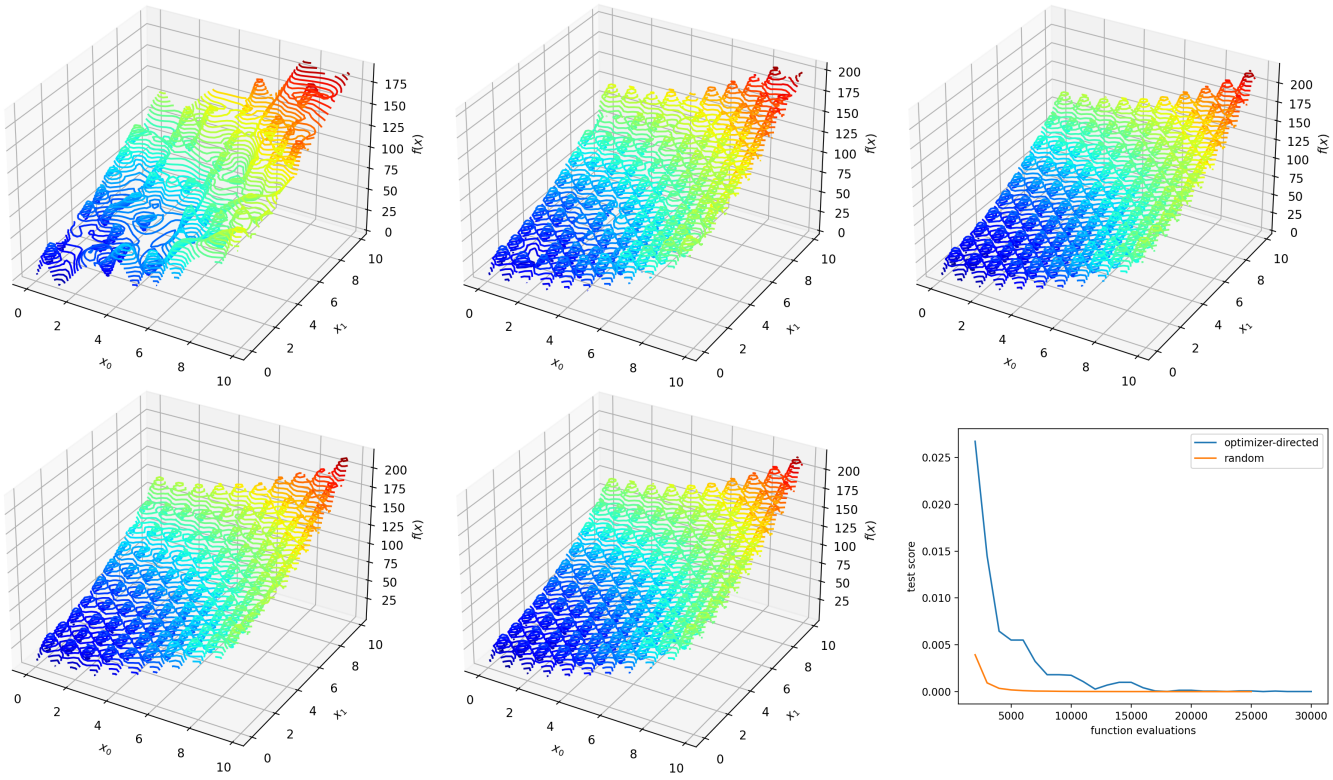


Figure 2. 2D Rastrigin function. Surrogates are plotted with inputs $x = (x_0, x_1)$ and output $z = f(x)$. Candidate surrogates learned with a thin-plate RBF estimator using "sparsity" sampling, a "loose" tolerance, and a test metric for validity based on the average graphical distance between the learned surrogate and sampled data. Top row: Sampling using ensembles of 16 optimizers, after the initial, tenth, and final iteration. The final surrogate is visually identical to truth, and the surrogate reproduces all local extrema within the desired accuracy. Bottom row: Sampling using ensembles of 500 points, after the initial and tenth iteration. Bottom row, right: test score per sample. Note that the test score for pure systematic random sampling converges faster than optimizer-directed sampling, as may be expected for a metric based on the average surrogate misfit.

Function	ndim	Random		Optimizer-directed	
		loose	strict	loose	strict
Easom	2	2000	8000	2939	17967
Rosenbrock	2	2000	7000	7317	12111
Rastrigin	2	7000	25000	18579	32308
Michalewicz	2	30000	–	30696	–
Hartmann	6	11000	–	26411	–
Rosenbrock	8	15000	–	31487	–

(a)

Function	ndim	bounds	Optimizer-directed	
			Powell	Nelder-Mead
Ackley	2	[-1,1]	3746	1631
Branins	2	[-10,20]	2767	1007
Rosenbrock	3	[-3, 3]	4796	1733
Michalewicz	5	[0, 3]	12745	1116
Hartmann	6	[-1, 1]	11896	1393
Rosenbrock	8	[-6, 6]	18430	1185

(b)

Table 1. (a) Number of evaluations required for several benchmark functions to reach reach tol_{stop} for both "loose" and "strict" tolerances, using a `SparsitySampler` with bounds $\mathbf{x} \in [0, 10]$. In all cases, systematic random sampling converges more quickly to a surrogate that is valid for all future data. Using optimizer-directed sampling, however, ensures that the function extrema are known. The optimizer used was a `NelderMeadSimplexSolver` at the default configuration. Less strict convergence requirements will reduce the number of evaluations required by the optimizer, potentially at the cost of some accuracy in the vicinity of the extrema. In (b) we present the number of function evaluations required to find a valid surrogate, where *converged* and *test* are defined as in Eq. (5), with $tol_{sum} = 10^{-3}$ and $tol_{max} = 10^{-6}$, and $train$, $data$, and $metric$. We used a `LatticeSampler` with an ensemble of 4 optimizers with the default configuration. The sampler is configured to run until all of the optimizers in the ensemble have terminated.

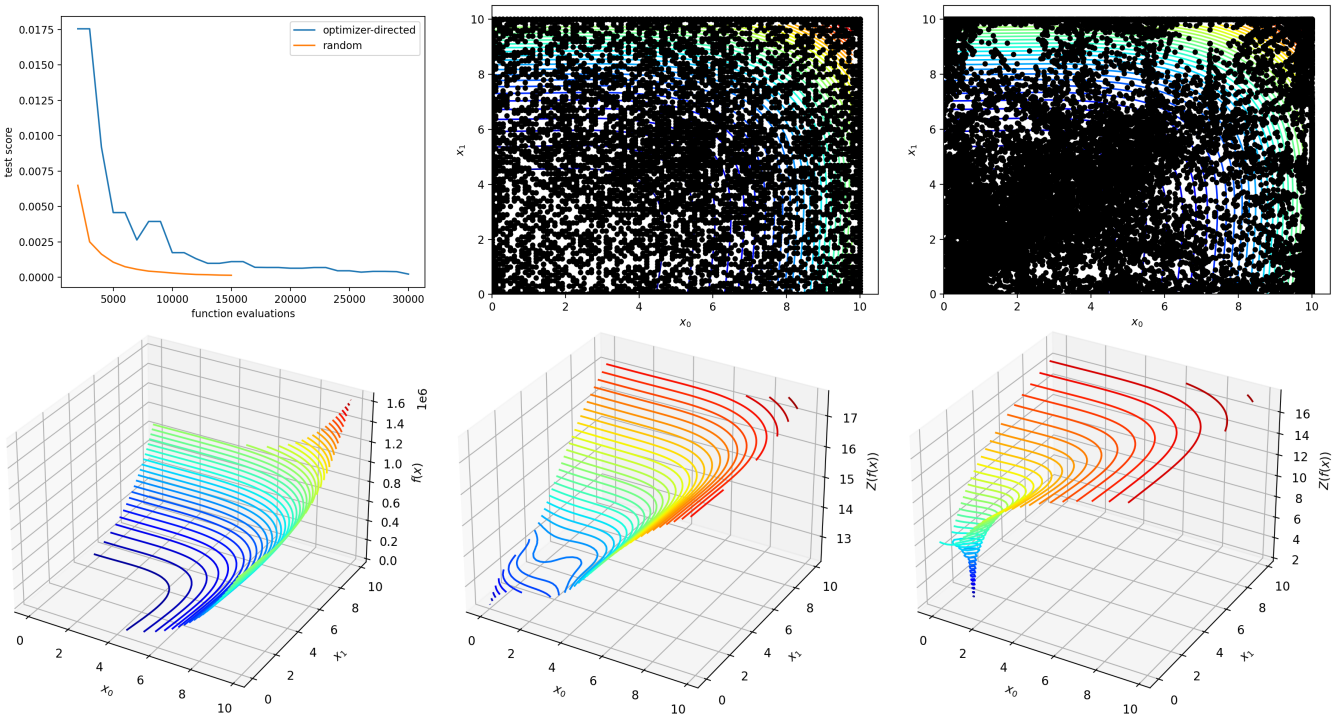
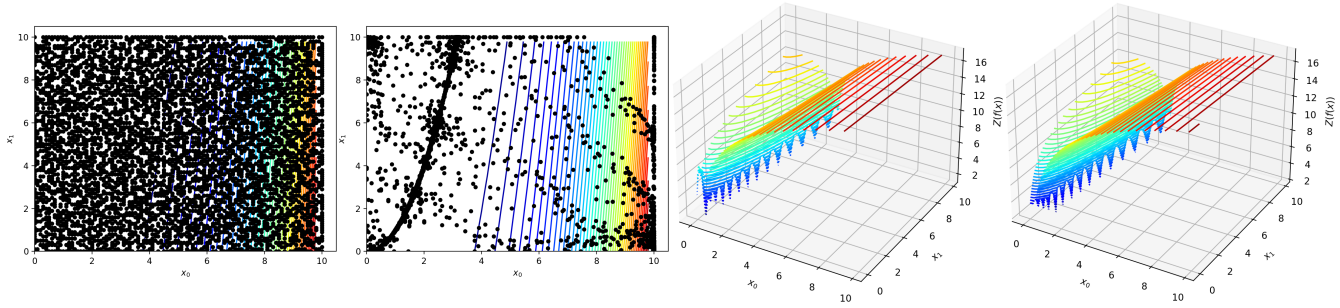
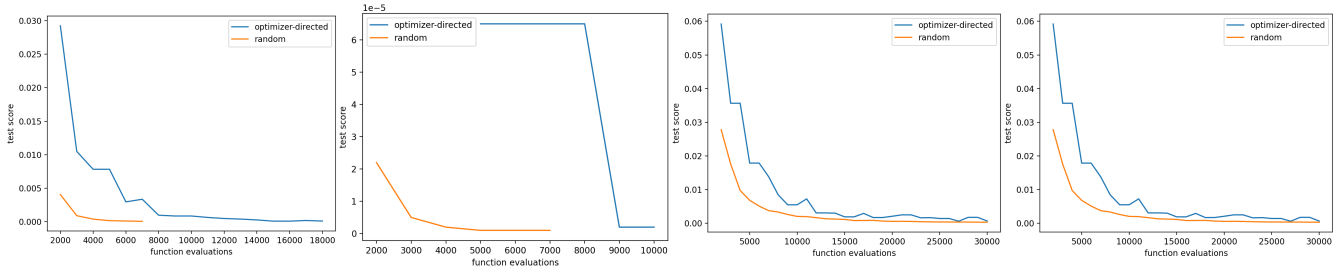


Figure 3. Similar to Fig. 2 but for the 8D Rosenbrock function. Top row, left: test score per sample. Top row, center: model evaluations sampled with the random sampling strategy. Top row, right: optimizer-directed sampling. Bottom row, left: surrogates produced with either sampling approach are visually identical to the truth. Bottom row, center: log-scaled view of surrogate from random sampling near the global minimum. Bottom row, right: log-scaled view of surrogate from optimizer-directed sampling near the global minimum, identical to the truth. Note that while pure systematic random sampling converges faster, optimizer-directed sampling provides a more accurate surrogate near the critical points.



(a) 2D Rosenbrock function. Surrogates are plotted with inputs $x = (x_0, x_1)$ and output $z = Z(f(x))$, where $Z = log(4 \cdot f(x) + 1) + 2$ scaling is used to better view the region around the global minimum. Candidate surrogates, learned with a thin-plate RBF estimator using "sparsity" sampling, a "loose" tolerance, and a test metric for validity based on the average graphical distance between the learned surrogate and sampled data. Left-right: model evaluations sampled with random sampling; optimizer-directed sampling; log-scaled view of surrogate from random sampling; and log-scaled view of surrogate from optimizer-directed sampling, which reproduces truth. Note that convergence occurs quickly using either strategy, where the *converged* condition is met with no more than two iterations.



(b) Convergence of *test* score versus model evaluations for different benchmark functions learned with a thin-plate RBF estimator using "sparsity" sampling, and a test metric for validity based on the average graphical distance between the learned surrogate and sampled data. Left-Right: 2D Rastrigin function with loose tolerance; 2D Rosenbrock function at strict tolerance; 2D Michalewicz function at loose tolerance; and 2D Michalewicz function at loose tolerance. The performance of either approach for the 2D Rosenbrock is as for the 8D function (Fig. 3). Both, random and optimizer-directed approaches, do equally poorly with the 2D Michalewicz.

Figure 4

to undergo a PT to a phase composed of deconfined quarks and gluons. This quark-hadron PT is studied experimentally in heavy-ion collisions on Earth and might be present in the interior of neutron stars[35]. However, there are large uncertainties regarding the critical temperatures and densities for the onset of the PT.

Numerical studies of matter during heavy-ion collisions, core-collapse supernovae, neutron star merger events and within the interior of neutron stars, require the usage of a nuclear EOS[36]. The most common approach to creating an EOS over a large density range is to select realistic hadronic and quark models and connect them either via a Maxwell or a Gibbs construction to describe the PT[37]. The Gibbs construction assumes that conservation laws are fulfilled globally in the quark-hadron mixed-phase, which results in pressure being a smooth function of the density. For the Maxwell construction, only baryon number is conserved globally, while other conservation laws like electric charge neutrality are fulfilled locally in the quark and hadronic phases.[37–39]. Neither of these constructions is currently ruled out, but the Maxwell construction usually leads to a more extreme behavior during the PT, leading to a pressure plateau in the so-called quark-hadron mixed phase.

For astrophysical simulations of core-collapse supernovae and neutron star mergers, nuclear matter EOS tables are used with thermodynamic quantities being functions of n_b, the proton fraction y_p and the temperature T[36]. The construction of these tables is time-intensive with many points in the (n_b, y_p, T) space, while their implementation in computational fluid dynamics (CFD) problems requires interpolation and inversion routines. A new approach to facilitate the usage of EOS tables is the construction of analytic fitting functions, which can then be included in the numerical simulations instead of an EOS table[40]. While promising, this method has yet to be shown to be able to accurately model extreme features like high-density PTs.

To demonstrate our framework, we implement a quark-hadron PT into a simple nucleonic model that is frequently used in astrophysics and nuclear physics[41,42]. Here, the nucleonic EOS is derived from the Skyrme-Hartree-Fock self-consistent mean-field model[43]. The energy of the system is determined as the expectation value of an effective nuclear Hamiltonian, which contains the zero-range Skyrme nuclear interaction[44]. For high-density neutron star interiors at zero or low temperatures, nuclear matter can be treated as degenerate and infinite with constant density. This greatly simplifies the expression for the Skyrme energy density functional. In addition, the many-body state of the system can be expressed as a Slater determinant of uncorrelated plane wave states from lowest momentum up to the Fermi momentum. As a result, the energy per baryon of nuclear matter composed of neutrons and protons with densities ρ_n and ρ_p, respectively, can be written in some an analytic form. The energy per baryon of nuclear matter composed of neutrons and protons with densities ρ_n and ρ_p are given, respectively, by:

$$\begin{aligned}\frac{E}{A}(y_p,\rho) = {} & \frac{3}{5}\left(\frac{\hbar^2}{2m}\rho\right)^{\frac{2}{3}} F_{5/3} \\ & + \frac{1}{8}t_0\rho\left[2(x_0+2)-(2x_0+1)F_2\right] \\ & + \frac{1}{48}t_3\rho^{\alpha+1}\left[2(x_3+2)-(2x_3+1)F_2\right] \\ & + \frac{3}{40}\left(\frac{3\pi^2}{2}\right)^{\frac{2}{3}}\rho^{\frac{5}{3}}\left[\left[t_1(x_1+2)+t_2(x_2+1)\right]F_{\frac{5}{3}}\right. \\ & \left. + \frac{1}{2}\left[t_2(2x_2+1)-t_1(2x_1+1)\right]F_{\frac{8}{3}}\right], \qquad (6)\end{aligned}$$

$$F_m(y_p) = 2^{m-1}\left[y_p^m + (1-y_p)^m\right], \qquad (7)$$

$$\rho = \rho_n + \rho_p, \;\; y_p = \rho_p/\rho. \qquad (8)$$

The parameters $x_{1\ldots3}$, $t_{1\ldots3}$ and α are fitted to reproduce properties of nuclei, such as binding energy, or of neutron stars, such as the neutron star radii.

Given the energy or energy per baryon, other thermodynamic properties, such as pressure, can then be determined by standard relations[43]. The PT is modeled by the Maxwell construction, while quark matter is described by the MIT Bag model, where quarks are non-interacting fermions with a negative confinement pressure, the so-called bag constant[39,45]. The latter model ensures that quarks are confined into neutrons and protons at low densities. Quark matter in our approach uses a bag constant of 170 MeV. It is composed of up, down, and strange quarks, where we assume that the masses of up and down quarks are negligible in comparison to their chemical potential, while the strange quark mass is set to 150 MeV. Although nucleonic and quark matter are given by simple models, our intention here is to demonstrate the ability of our framework to use expensive EOS data, even in the presence of a PT, to find a surrogate that can be directly used in high-fidelity CFD codes. EOS tables in astrophysical simulations usually have three degrees of freedom: n_b, y_p and T[36], but for simplicity, we will model a system where the pressure is a function of n_b and y_p only, and temperature effects are negligible, which is a reasonable assumption for systems such as neutron stars interiors.

We use our approach to find an accurate surrogate for the quark-hadron EOS for $0.04\,\mathrm{fm}^{-3} \leq n_b \leq 1.6\,\mathrm{fm}^{-3}$ and $0 \leq y_p \leq 0.6$. We used "lattice" sampling with an ensemble of 40 Nelder-Mead solvers at the default configuration, and a surrogate learned using a thin-plate RBF. Here, we defined *test* validity as in Eq. (5) with $tol_{max} = 10^{-6}$ and $tol_{sum} = 10^{-3}$, and *train*, *converged*, *data*, and *metric* as defined in Section . The results are plotted in Figure 5(a), which shows the entire n_b-y_p plane. The pressure plateau of the PT is clearly visible with critical density for the onset of the mixed phase moving to higher values for increasing proton fraction as discussed in[39]. Figure 5(b) gives a more detailed view of the EOS by showing the pressure profiles for fixed n_b. It can be seen from the figure that no systematic errors arise in the predicted values

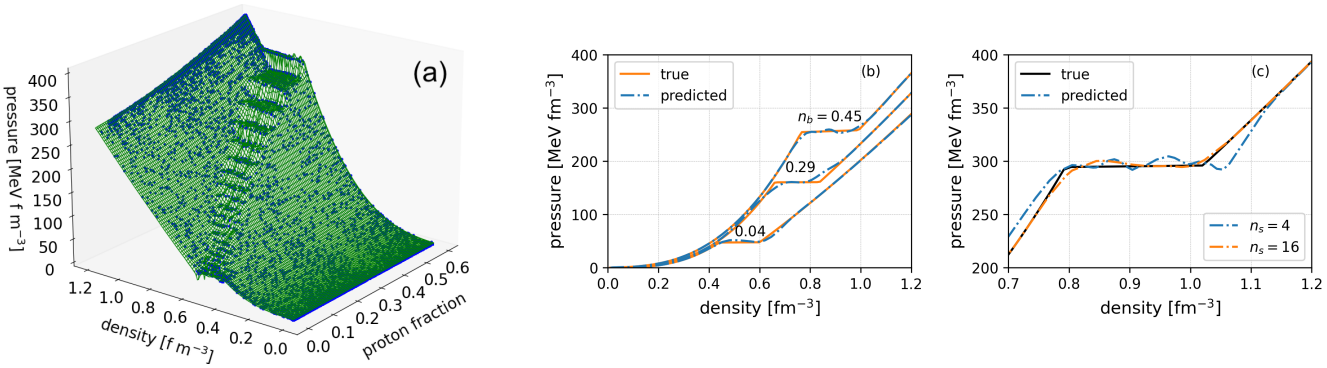


Figure 5. (a) Simulations (dots) and predicted values (surface) of the equation of state for quark matter. The initial search domain chosen as density $[fm^{-3}] \in [0.04, 1.2]$ and proton fraction $\in [0., 0.6]$ was sampled with a lattice sampler. We used 40 Nelder-Mead solvers and *test* validity defined as in Eq. (5) with $tol_{max} = 10^{-6}$, $tol_{sum} = 10^{-3}$, and *train*, *converged*, *data*, and *metric* as defined in Section . A valid surrogate for the EOS that correctly describes the plateau region was found after 7382 function evaluations. (b) Quark matter pressure as function of density for a given proton fraction. (c) Quark matter pressure as a function of density for a given proton fraction and for various number of optimizers. The orange curve shows the results from more expensive simulation and the blue curve represents prediction using our methodology. Note as we increase the size of the ensemble of solvers (n_s) directed to find the critical points, the accuracy of the surrogate obtained from a single learning step using a thin-plate RBF improves.

of the pressure with either proton fraction or density. Note that the RBFs will reproduce the phase transition accurately if enough of the inflection points of the response surface are sampled. As the optimizers discover all the critical points in the surface (here, the points in the discontinuity), the smooth functions also begin to reproduce these discontinuities. We illustrated this point in Fig. 5(c).

Discussion

We presented an online learning strategy that is designed to produce valid surrogates for a chosen quality metric. The approach works well in generating surrogates for existing data and can also be applied with regard to future data. We demonstrated an application of online learning where the selection of training data is done with a sampling strategy and an iterative approach is applied to improve surrogate validity versus the chosen metric. We gave evidence that if the critical points of the model's response surface are known, a robust estimator (e.g. thin-plate RBF interpolation or a MLP neural network) should be able to create a surrogate that reproduces the behavior of more expensive model exactly. We presented an optimizer-directed sampling strategy that is effective at sampling the critical points of a model's response surface. We then compared the efficiency of different sampling strategies in learning surrogates that are valid for benchmark functions, even in the presence of newly sampled data. Note that if the surrogate was found to be invalid for newly acquired data, our online approach can be used to improve it iteratively.

For selected benchmark functions, we used a sparse sampling approach that produced new draws at the least-populated points in parameter space. We compared this to an optimizer-directed approach where each initial draw is used as starting point for an optimizer that runs to termination. At first blush, it seemed that the traditional sampling strategy outperformed the optimizer-directed one for all benchmark functions (see Table 1a). However, we used a metric that was based on the error in the surrogate's predicted value versus truth, averaged out over the entire response surface. Hence, it should not be surprising that our methodology produced surrogates that are, on average, of high quality across the entire parameter range. One might conclude that a traditional sampling strategy, especially one that provides more diffuse sampling than an optimizer-directed strategy, is more efficient at generating valid surrogates when the average misfit across parameter space measures the validity. However, we note that the default optimizer configuration was used in testing the efficiency of the sampling strategy, and *tuning* the optimizer may make a substantial difference in the efficiency of the optimizer-directed strategy.

We tested two different optimizers and determined their convergence behavior using the default settings. We found that the choice of optimizer can affect the efficiency by over an order of magnitude (see Table 1b), and that the use of stronger termination conditions can also impact the efficiency by a similar amount . With that, given that one does some upfront work to tune the optimizer for a given problem, the optimizer-directed approach can *easily* be more efficient than a traditional sampling approach.

Importantly, we also noted that our applied metric made no guarantee regarding the quality of the surrogate *in the neighborhood of the critical points*. Returning to our conjecture, the finding of a response function's critical points is key to guarantee the long-term validity of the surrogate as new data is collected. We found that an optimizer-directed approach is

superior at minimizing the model error in the neighborhood of the critical points (see Figure 3), even when the metric does not call for that explicitly. Conversely, a traditional sampling strategy is blind to the response surface and demonstrated a much larger misfit near the critical points. Thus, using a metric that judges the quality of the surrogate by the misfit at the critical points should produce high-quality surrogates with an optimizer-directed approach with even greater efficiency.

For a physical system, the critical points of a response surface are usually associated with the occurrence of new phenomena. This provides additional motivation to reduce the misfit near the critical points as much as possible. With that, we applied our methodology to two physics test problems. We showed that we could efficiently learn surrogates for equation-of-state calculations of dense nuclear matter, yielding excellent agreement between the surrogate and model for a wide parameter range and in the region that includes a phase transition. We also showed that our methodology can produce highly-accurate surrogates for radial distribution functions from expensive Molecular Dynamics simulations for neutral and charged systems of several dimensions and across an extensive range of thermodynamic conditions (see Supplementary information). While our demonstrations were focused on two specific problems, the methodology and associated code are agnostic to the domain science and can be utilized for a wide variety of physics scenarios.

A standard metric that is used to determine the validity of a surrogate is the model error, given in Eq. (11). This definition assesses the quality of the surrogate by measuring its distance from the observed data. Unfortunately, for a small set of observed data that is not representative, any learned surrogate will likely become invalidated with the addition of new data. A potentially more robust assessment of model quality considers training a surrogate with a statistical metric, such as the *expected* model error. It can be defined to take into account any knowledge about the data-generating distributions (for input and output values) and any uncertainty in the input and output parameters of the model. Complex real-world models are often non-deterministic; thus, an appropriate goal is to either find a surrogate that is guaranteed to be accurate under uncertainty or a surrogate that is guaranteed to be robust under uncertainty. With some minor adjustments, such as adding a strategy to timestamp or invalidate training data, our methodology can be leveraged to build and maintain accurate surrogates for time-dependent models. In future work, we will apply our methodology to produce surrogates that are guaranteed to be either accurate or robust under uncertainty and similarly demonstrate the ability to guarantee the accuracy of surrogates for time-dependent models.

The code implemented for our methodology facilitates saving the learned surrogates to a DB, where they can be easily utilized within coarse-grain calculations and codes, as in[15]. Similarly, any sampled data used in this work is seamlessly saved to a DB.

Methods

Surrogate Validity.
Our general procedure to create a valid surrogate for an expensive model is shown in Fig. 1. The steps are iterative and include explicit validation and update mechanisms. To simplify computational complexity, we first link the model to a DB. Thus, when the model is evaluated, its inputs and output are automatically stored. The DB of model evaluations is used later to train candidate surrogates. The corresponding surrogate is retrieved from the surrogate DB and tested for validity during model evaluation. If no stored surrogate exists, then we skip testing and proceed directly to learning a candidate surrogate. Validity is defined as

$$\textit{test}(\Delta) \text{ is true} \tag{9}$$

where *test* is a function of the graphical distance, Δ

$$\begin{aligned} \Delta_y &= \inf_{\mathbf{x}\in\mathscr{X}} \left| \hat{y}(\mathbf{x}|\xi) - y \right| + \Delta_x, \\ \Delta_x &= \left| \mathbf{x} - \mathbf{x}' \right| \text{ or } 0 \end{aligned} \tag{10}$$

with $(\mathbf{x}', y)$ a point in the DB of model evaluations, and $\mathscr{X}$ the set of all valid inputs $\mathbf{x}$ for the surrogate $\hat{y}$ with hyperparameters ξ. Δ_x and Δ_y are the pointwise Δ for $(\mathbf{x}', y)$. If $\Delta_x = 0$, we ignore the distance of the inputs while Δ_y is the minimum vertical distance of point y from the surrogate.

If Eq. (9) deems the surrogate to be valid, the execution stops. Otherwise, we update the surrogate by training against the DB of stored model evaluations. We define validity when training a surrogate similar to Eq. (9), but with the function *train* replacing *test*. We train the surrogate in terms of a quality metric, which is typically a distance such as $\delta = \sum_y \Delta_y$, or more generally

$$\delta = \textit{metric}(\hat{y}(\mathbf{x}|\xi),\ \textit{data}) \tag{11}$$

with *metric* being a distance function between the surrogate and all model evaluations, *data*. If after training a surrogate has a smaller δ compared to the current best surrogate, then we store the updated surrogate in the surrogate DB and continue to improve the surrogate until *train* is satisfied. In the case that training ultimately fails to produce a valid surrogate, we use a sampler to generate model evaluations at new $(\mathbf{x}', y)$ and the process restarts.

Our general procedure for producing a valid surrogate is extended for asymptotic validity by adding a validity convergence condition

$$\textit{converged}(\Delta) \text{ is true} \tag{12}$$

to be called after the surrogate is deemed *test* valid, as in Eq. (9). Thus, instead of stopping execution when the surrogate is *test* valid, the latter merely completes an iteration. If not *converged*, we trigger a new iteration by sampling new data and continue to iterate until the surrogate validity has *converged*. This iterative procedure is more likely to generate a surrogate that is valid for all future data when Eq. (12)

requires some form of convergence behavior for *test* over several iterations. When the DB of model evaluations is sparsely populated, we expect that any new data will likely trigger a surrogate update.

Learning Strategy.
Our procedure is online, as a sampler can request new model evaluations on-the-fly, which populate to a DB, and our surrogate is updated by querying the DB and training on the stored model evaluations. Online learning is greatly facilitated by automation of the learning process. Our general procedure for automating the production of a valid surrogate is shown in Fig. 1, and is extended to asymptotic validity. As mentioned earlier, we will use a RBF to generate our surrogates, where we leverage *mystic* for the automation and quality assurance of surrogate production.

Let us assume $y(\mathbf{x})$ is an arbitrary function of vector $\mathbf{x}$ represented on a subset of $\mathbb{R}^n$, and that the value of y at input vectors $\mathbf{x}^j$ $(j = 1,\ldots,N)$ are the known *data* points stored in a DB of model evaluations. We seek to find a surrogate $\hat{y}(\mathbf{x})$ with the lowest possible number of the evaluations[46–48] satisfying Eq. (11). We use Eq. (11), with $\mathbf{x}' = \mathbf{x}$, as opposed to

$$\hat{y}(\mathbf{x}^j) = y(\mathbf{x}^j) \text{ for all } j = 1,\ldots,N \tag{13}$$

as we allow our interpolated surrogates to deviate from the data slightly, due to the use of `smooth` and `noise`. Using a RBF $\phi(r)$, the interpolated function can be written as:

$$\hat{y}(\mathbf{x}) = \sum_{j=1}^{N} \beta_j \phi(d(\mathbf{x},\mathbf{x}^j)), \tag{14}$$

where β_j are coefficients to be determined, and $d(\mathbf{x},\mathbf{x}^j)$ is a distance function similar to Eq. (11). If we choose $d(\mathbf{x},\mathbf{x}^j) = ||\mathbf{x}-\mathbf{x}^j||$ as the Euclidean distance between an arbitrary vector $\mathbf{x}$ and $\mathbf{x}^j$, the values of the coefficient vector $\beta = [\beta_1,\beta_2,\cdots,\beta_N]^T$ are determined by solving the linear system, $\mathbf{M}\beta = \mathbf{Y}$, where $\mathbf{M}$ is an $N \times N$ symmetric matrix with elements $M_{ij} = \phi(||\mathbf{x}^i-\mathbf{x}^j||)$, and $\mathbf{Y} = [y(\mathbf{x}^1),y(\mathbf{x}^2),\ldots,y(\mathbf{x}^N)]^T$. In this work, we use `thin_plate` ($\phi = r^2\ln(r)$) RBF to interpolate the data. To prevent issues due to singular matrix $\mathbf{M}$, and to provide some randomness in each learned surrogate, we add a very small amount of Gaussian noise to the input data.

Sampling Strategy.
Sampling is an integral part of our online learning workflow and is used to generate new data points $(\mathbf{x}',y)$ that help inform the learning algorithm whenever training fails to produce a valid surrogate. As the goal is an asymptotically valid surrogate, we also use sampling to kick-start a new iteration after Eq. (9) deems the current iteration's surrogate to be valid (see Fig. 1). While our workflow's sampling and learning components are fundamentally independent and can run asynchronously, they are linked through the DB of stored model evaluations. The data points generated by the sampler are populated to the DB, while the learning algorithm always uses the data contained in the DB when new training is requested. If there were no concerns about minimizing the number of model evaluations, we could have samplers run continuously, feeding model evaluations into the DB. However, as described above, we explicitly include sampling as part of the iterative workflow to minimize the number of model evaluations.

We conjectured that (given the training data) a learned surrogate which, at a minimum, includes all of the critical points of a response surface $y(\mathbf{x})$ is guaranteed to be valid for all future data. Thus, we postulate that a sampling strategy that uses *optimizer-directed* sampling will be most efficient in discovering all the critical points of $y(\mathbf{x})$. We distinguish *optimizer-directed* sampling from *traditional* sampling. Optimizer-directed sampling uses an optimizer to direct the sampling toward a goal. In contrast, traditional methods, such as simple random sampling generally ignore the response of the function $y(\mathbf{x})$. The utility of simple random sampling is that all the samples will draw (with replacement) from a distribution, and thus all sample points can be chosen simultaneously. Subsequently, $y(\mathbf{x})$ can be evaluated in parallel for all points drawn in the sampling. An optimizer-directed approach uses traditional sampling to generate samples for the first draw, then uses each first draw member as a starting point for an optimizer that will direct the sampling of the second and subsequent draws toward a critical point on the response surface. When an optimizer's termination condition is met, traditional sampling is again used to generate a new starting point for a new optimizer, which then proceeds to termination as above. Thus, while an optimizer-directed strategy may be less efficient in generating new data points, it should be more efficient at finding the critical points of the response surface, and thus be the preferred strategy when a surrogate is required to be asymptotically valid.

1 Code availability

The code, as well as the sampled data and learned surrogates, relevant to this work are available on Code Ocean.

2 Acknowledgments

Research presented in this article was supported by Los Alamos National Laboratory under the Laboratory Directed Research and Development program (project numbers 20190005DR, 20200410DI, and 20210116DR), by the Department of Energy Advanced Simulation and Computing under the Beyond Moore's Law Program, and by the Uncertainty Quantification Foundation under the Statistical Learning program. Los Alamos National Laboratory is operated by Triad National Security, LLC, for the National Nuclear Security Administration of U.S. Department of Energy (Contract No. 89233218CNA000001). The Uncertainty Quantification Foundation is a nonprofit dedicated to the advancement of predictive science through research, education, and the development and dissemination of advanced technologies. The

authors would like to thank Jeff Haack for his very useful feedback on the manuscript. This document is LA-UR-20-24947.

3 Author contributions statement

A.D., M.M., and M.S.M. conceived the project. M.M developed the software. A.D., M.M. and I.S. performed simulations and prepared figures. All authors were responsible for the formal analysis.

4 Additional information

Competing financial interests: The authors declared no competing financial interests.

References

1. Coveney, P. V., Boon, J. P. & Succi, S. Bridging the gaps at the physics-chemistry-biology interface. *Philos. Transactions Royal Soc. Lond. Ser. A* **374**, 20160335 (2016).
2. Paxton, B. *et al.* Modules for Experiments in Stellar Astrophysics (MESA): Convective Boundaries, Element Diffusion, and Massive Star Explosions. *ApJS* **234**, 34 (2018).
3. Stanton, L. G., Glosli, J. N. & Murillo, M. S. Multiscale molecular dynamics model for heterogeneous charged systems. *Phys. Rev. X* **8**, 021044 (2018).
4. Kress, J. D., Cohen, J. S., Horner, D. A., Lambert, F. & Collins, L. A. Viscosity and mutual diffusion of deuterium-tritium mixtures in the warm-dense-matter regime. *Phys. Rev. E* **82**, 036404 (2010).
5. Brown, E. W., Clark, B. K., DuBois, J. L. & Ceperley, D. M. Path-integral monte carlo simulation of the warm dense homogeneous electron gas. *Phys. Rev. Lett.* **110**, 146405 (2013).
6. Dornheim, T. *et al.* The static local field correction of the warm dense electron gas: An ab initio path integral Monte Carlo study and machine learning representation. *J. Chem. Phys.* **151**, 194104 (2019).
7. Schmidt, J., Marques, M., Botti, S. & Marques, M. Recent advances and applications of machine learning in solid-state materials science. *npj Comput. Mater.* **5** (2019).
8. Liu, Y., Zhao, T., Ju, W. & Shi, S. Materials discovery and design using machine learning. *J. Materiomics* **3** (2017).
9. Barros, V. *et al.* (eds.) *Climate Change 2014 Impacts, Adaptation, and Vulnerability* (Cambridge University Press, New York, 2014).
10. Wigley, P. *et al.* Fast machine-learning online optimization of ultra-cold-atom experiments. *Sci. Reports* **6** (2016).
11. Scheinker, A. & Gessner, S. Adaptive method for electron bunch profile prediction. *Phys. Rev. Accel. Beams* **18** (2015).
12. Noack, M. *et al.* A kriging-based approach to autonomous experimentation with applications to x-ray scattering. *Sci. Reports* **9** (2019).
13. Lubbers, N. *et al.* Modeling and scale-bridging using machine learning: nanoconfinement effects in porous media. *Sci. Reports* **10**, 13312 (2020).
14. Diaw, A. *et al.* Multiscale simulation of plasma flows using active learning. *Phys. Rev. E* **102**, 023310 (2020).
15. Roehm, D. *et al.* Distributed Database Kriging for Adaptive Sampling (D^2 KAS). *Comput. Phys. Commun.* **192**, 138–147 (2015).
16. Coulomb, J.-L., Kobetski, A., Caldora Costa, M., Maréchal, Y. & Jonsson, U. Comparison of radial basis function approximation techniques. *COMPEL-The international journal for computation mathematics electrical electronic engineering* **22**, 616–629 (2003).
17. Park, J. & Sandberg, I. W. Universal approximation using radial-basis-function networks. *Neural computation* **3**, 246–257 (1991).
18. Wu, Y., Wang, H., Zhang, B. & Du, K.-L. Using radial basis function networks for function approximation and classification. *ISRN Appl. Math.* **2012** (2012).
19. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, 318–362 (MIT press Cambridge, MA, 1986).
20. McKerns, M., Hung, P. & Aivazis, M. mystic: highly-constrained non-convex optimization and UQ (2009). http://pypi.python.org/pypi/mystic.
21. McKerns, M., Strand, L., Sullivan, T. J., Fang, A. & Aivazis, M. Building a framework for predictive science. In *Proceedings of the 10th Python in Science Conference*, 67–78 (2011). http://arxiv.org/pdf/1202.1056.
22. Khoshnoud, F., Esat, I. I., de Silva, C. W., McKerns, M. M. & Owhadi, H. Self-Powered Dynamic Systems in the Framework of Optimal Uncertainty Quantification. *J. Dyn. Syst. Meas. Control.* **139** (2017).
23. Owhadi, H., Scovel, C., Sullivan, T. J., McKerns, M. & Ortiz, M. Optimal Uncertainty Quantification. *SIAM Rev.* **55**, 271–345 (2013).
24. Sullivan, T. J. *et al.* Optimal uncertainty quantification for legacy data observations of Lipschitz functions. *ESAIM Math. Model. Numer. Anal.* **47**, 1657–1689 (2013).
25. Kamga, P.-H. T. *et al.* Optimal uncertainty quantification with model uncertainty and legacy data. *J. Mech. Phys. Solids* **72**, 1–19 (2014).

26. Li, C. W., McKerns, M. M. & Fultz, B. A raman spectrometry study of phonon anharmonicity of zirconia at elevated temperatures. *J. Am. Ceram. Soc.* **94**, 224–229 (2011).

27. Belak, J., Orikowski, D., Applegate, S., Owhadi, H. & McKerns, M. Quantifying model uncertainty (2012). LLNL-PRES-585774.

28. McKerns, M., Alexander, F., Hickmann, K., Sullivan, T. & Vaughn, D. Optimal Bounds on Nonlinear Partial Differential Equations in Model Certification, Validation, and Experiment Design (2020). https://arxiv.org/abs/2009.06626.

29. McKerns, M., Roth, L., Iyengar, N. & Lamm, D. Rigorous bounds on the failure of shielding due to helium-ion radiation (2021). In preparation.

30. Biwer, C., Vogel, S., McKerns, M. & Ahrens, J. Spotlight: Distributed-computing for rietvled analyses using an ensemble of local optimizers (2019). http://github.com/lanl/spotlight.

31. Rastrigin, L. A. *Systems of External Control* (Mir Publishers, Moscow, 1974). (in Russian).

32. Rosenbrock, H. An automatic method for finding the greatest or least value of a function. *The Comput. J.* **3**, 175–184 (1960).

33. Lonardoni, D., Tews, I., Gandolfi, S. & Carlson, J. Nuclear and neutron-star matter from local chiral interactions. *Phys. Rev. Res.* **2**, 022033, DOI: 10.1103/PhysRevResearch.2.022033 (2020).

34. Annala, E., Gorda, T., Kurkela, A., Nättilä, J. & Vuorinen, A. Evidence for quark-matter cores in massive neutron stars. *Nat. Phys.* DOI: 10.1038/s41567-020-0914-9 (2020).

35. Dexheimer, V. Tabulated neutron star equations of state modelled within the chiral mean field model. *Publ. Astron. Soc. Aust.* **34**, DOI: 10.1017/pasa.2017.61 (2017).

36. Typel, S., Oertel, M. & Klähn, T. CompOSE CompStar online supernova equations of state harmonising the concert of nuclear physics and astrophysics compose.obspm.fr. *Phys. Part. Nucl.* **46**, 633–664, DOI: 10.1134/S1063779615040061 (2015).

37. Hempel, M., Pagliara, G. & Schaffner-Bielich, J. Conditions for phase equilibrium in supernovae, protoneutron, and neutron stars. *Phys. Rev. D* **80**, 125014, DOI: 10.1103/PhysRevD.80.125014 (2009).

38. Glendenning, N. K. *Compact Stars: Nuclear Physics, Particle Physics and General Relativity*. Astronomy and Astrophysics Library (Springer New York, 1997).

39. Fischer, T. *et al.* Core-collapse supernova explosions triggered by a quark-hadron phase transition during the early post-bounce phase. *The Astrophys. J. Suppl. Ser.* **194**, 39, DOI: 10.1088/0067-0049/194/2/39 (2011).

40. Raithel, C. A., Özel, F. & Psaltis, D. Finite-temperature Extension for Cold Neutron Star Equations of State. *Astrophys. J.* **875**, 12, DOI: 10.3847/1538-4357/ab08ea (2019).

41. Chabanat, E., Bonche, P., Haensel, P., Meyer, J. & Schaeffer, R. A skyrme parametrization from subnuclear to neutron star densities part ii. nuclei far from stabilities. *Nucl. Phys. A* **635**, 231 – 256, DOI: https://doi.org/10.1016/S0375-9474(98)00180-8 (1998).

42. Schneider, A. S., Roberts, L. F. & Ott, C. D. Open-source nuclear equation of state framework based on the liquid-drop model with skyrme interaction. *Phys. Rev. C* **96**, DOI: 10.1103/physrevc.96.065802 (2017).

43. Stone, J. & Reinhard, P.-G. The skyrme interaction in finite nuclei and nuclear matter. *Prog. Part. Nucl. Phys.* **58**, 587 – 657, DOI: https://doi.org/10.1016/j.ppnp.2006.07.001 (2007).

44. Skyrme, T. The effective nuclear potential. *Nucl. Phys.* **9**, 615 – 634, DOI: https://doi.org/10.1016/0029-5582(58)90345-6 (1958).

45. Chodos, A., Jaffe, R. L., Johnson, K., Thorn, C. B. & Weisskopf, V. F. New extended model of hadrons. *Phys. Rev. D* **9**, 3471–3495, DOI: 10.1103/PhysRevD.9.3471 (1974).

46. Schaback, R. & Wendland, H. Adaptive greedy techniques for approximate solution of large RBF systems. *Numer. Algorithms* **24**, 239–254 (2000).

47. Rocha, H. On the selection of the most adequate radial basis function. *Appl. Math. Model.* **33**, 1573 – 1583 (2009).

48. Dorvlo, A. S., Jervase, J. A. & Al-Lawati, A. Solar radiation estimation using artificial neural networks. *Appl. Energy* **71**, 307 – 319 (2002).