



PAPER • OPEN ACCESS

## Deep energy-pressure regression for a thermodynamically consistent EOS model

To cite this article: Dayou Yu *et al* 2024 *Mach. Learn.: Sci. Technol.* **5** 015031

View the [article online](#) for updates and enhancements.

You may also like

- [Cooling Timescale for Protoneutron Stars and Properties of Nuclear Matter: Effective Mass and Symmetry Energy at High Densities](#)  
Ken'ichiro Nakazato and Hideyuki Suzuki
- [Observing Supernova Neutrino Light Curves with Super-Kamiokande. II. Impact of the Nuclear Equation of State](#)  
Ken'ichiro Nakazato, Fumi Nakanishi, Masayuki Harada et al.
- [Gravitational waves, neutrino emissions and effects of hyperons in binary neutron star mergers](#)  
Kenta Kiuchi, Yuichiro Sekiguchi, Koutarou Kyutoku et al.



## PAPER

## OPEN ACCESS

RECEIVED  
8 April 2023REVISED  
13 January 2024ACCEPTED FOR PUBLICATION  
5 February 2024PUBLISHED  
20 February 2024

Original Content from  
this work may be used  
under the terms of the  
[Creative Commons  
Attribution 4.0 licence](#).

Any further distribution  
of this work must  
maintain attribution to  
the author(s) and the title  
of the work, journal  
citation and DOI.



# Deep energy-pressure regression for a thermodynamically consistent EOS model

Dayou Yu<sup>1</sup> , Deep Shankar Pandey<sup>1</sup> , Joshua Hinz<sup>2</sup>, Deyan Mihaylov<sup>2</sup> , Valentin V Karasiev<sup>2</sup> ,  
S X Hu<sup>2</sup> and Qi Yu<sup>1,\*</sup>

<sup>1</sup> Rochester Institute Of Technology, Rochester, NY, United States of America

<sup>2</sup> Laboratory for Laser Energetics, University of Rochester, Rochester, NY, United States of America

\* Author to whom any correspondence should be addressed.

E-mail: [qi.yu@rit.edu](mailto:qi.yu@rit.edu)

**Keywords:** equation-of-state, meta-learning, uncertainty quantification

## Abstract

In this paper, we aim to explore novel machine learning (ML) techniques to facilitate and accelerate the construction of universal equation-Of-State (EOS) models with a high accuracy while ensuring important thermodynamic consistency. When applying ML to fit a universal EOS model, there are two key requirements: (1) a high prediction accuracy to ensure precise estimation of relevant physics properties and (2) physical interpretability to support important physics-related downstream applications. We first identify a set of fundamental challenges from the accuracy perspective, including an extremely wide range of input/output space and highly sparse training data. We demonstrate that while a neural network (NN) model may fit the EOS data well, the black-box nature makes it difficult to provide physically interpretable results, leading to weak accountability of prediction results outside the training range and lack of guarantee to meet important thermodynamic consistency constraints. To this end, we propose a principled deep regression model that can be trained following a meta-learning style to predict the desired quantities with a high accuracy using scarce training data. We further introduce a uniquely designed kernel-based regularizer for accurate uncertainty quantification. An ensemble technique is leveraged to battle model overfitting with improved prediction stability. Auto-differentiation is conducted to verify that necessary thermodynamic consistency conditions are maintained. Our evaluation results show an excellent fit of the EOS table and the predicted values are ready to use for important physics-related tasks.

## 1. Introduction

**Background.** Improving our understanding of high-energy-density physics and advancing research in the important fields of inertial confinement fusion (ICF) and planetary science relies on accurate equation-of-state (EOS) models, which cover a wide range of thermodynamic conditions [1, 2]. As technology is improving, experimental measurements of EOS are accessing higher and higher density and temperature conditions, such as those encountered in imploding ICF targets [3, 4] and white dwarfs [5]. Measurements at such conditions are extremely difficult to obtain and data are therefore sparse and mainly serve as benchmarks for the accuracy of theoretical models, which are also constantly evolving in both accuracy and range of conditions covered [6].

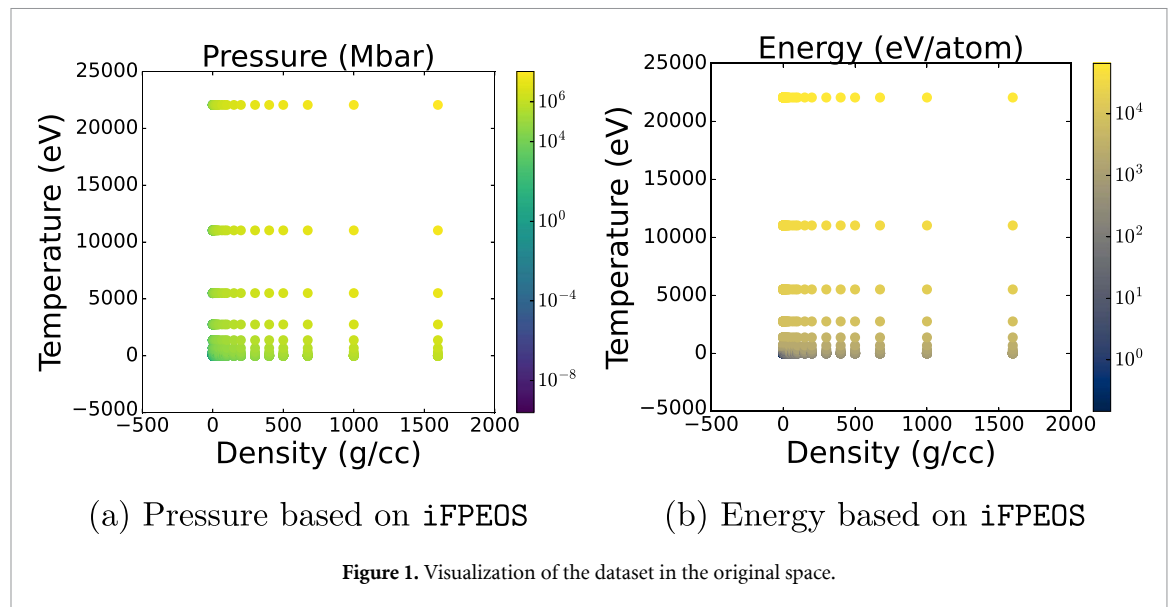
Recently, an improved version of the first-principles EOS table (iFPEOS) for deuterium is published [7], which is an update on FPEOS [8, 9] based improved theoretical methods such as *ab initio* molecular dynamics (AIMD). These methods are driven by density functional theory (DFT), where advanced meta-generalized gradient approximation (meta-GGA) exchange-correlation (XC) free energy density functional TSCANL [10], a high-accuracy non-interacting orbital-free free energy density functional LKTF $\gamma$ TF (see details in [7]) is used. Compared to previous models [11–15], iFPEOS showed better agreement with experimental measurements [4, 16–18] for temperatures  $T \sim 60\,000$  K and pressures up to

$P \sim 200$  GPa. However, for higher  $T - P$  regimes, iFPEOS fails to close the gap between latest theory and experiment. At such extreme conditions, both theoretical and experimental work can be considered in their pioneering stages and iFPEOS suggests that first-principles treatment beyond DFT might be necessary. Additionally, at such extreme conditions, first-principles simulations are challenging from the computational point of view as AIMD are significantly time-consuming and require a large allocation of computing resources. An AIMD run, corresponding to one point in iFPEOS, could take approximately a few hours to several days running on tens to hundreds of cores depending on thermodynamic conditions and methodology, with the low- $T$ , low-density orbital-based Kohn–Sham DFT calculations being much more computationally demanding than the high- $T$  orbital-free AIMD ones. Therefore, the immediate future of theoretical EOS models relevant to ICF faces two great challenges: (1) The high computational demands of a *single calculation* arising from the need to go beyond DFT; (2) The need for *many such calculations* in order to finely sample a wide range of thermodynamic conditions. As current developments in efficient *ab initio* algorithms and the performance of supercomputing clusters is relatively slow and incremental, one cannot overlook the potential of machine learning (ML) methods in the fast generation of dense and accurate EOS models from sparse data.

**ML based EOS modeling.** In recent years, ML has been increasingly adopted by the scientific research community to address the data-computation extensive challenges [19]. Given available observations, an ML model can be trained to learn the underlying patterns in the data, which can then be used to make a prediction at any density-temperature point of interest. A promising solution to build a universal EOS model is to leverage an iFPEOS table of finite size to train an ML model that can recover any missing values in the table and hence explain the wide range of behaviors of deuterium EOS. As some recent efforts in using ML to model EOS [20, 21] show, using a neural network (NN) surrogate model to provide EOS information is viable and provides advantages such as saving the memory cost of restoring all EOS tables, providing differentiability for downstream tasks, and accelerating simulations. Another important factor is as aforementioned, that an NN model can provide a universal approximation. This implies that an NN model can achieve any desired error rate on training. Plus, unlike interpolation methods which usually require neighborhood knowledge, a trained NN model can predict at any input point.

In our work, we propose novel extensions of a standard NN model to an encoder-decoder structure and improve the ML model design to address the unique challenges for modeling the EOS table. We first formally introduce these key challenges either identified by prior work or newly discovered by us. First, both the input and the output span a very wide range. As an example, figure 1 visualizes an iFPEOS table used in our experiments. The two input features, including density and temperature, cover a wide range that reaches  $10^{-6} \sim 10^3 \text{ g cm}^{-3}$  and  $10^{-5} \sim 10^5 \text{ eV}$ , respectively. Similarly, the two outputs, i.e. energy and pressure, span more than eight orders of magnitude. Meanwhile, the input-output dependency in certain ranges is highly sensitive, where a small change in the input may result in a significant variation in the output. This poses a major challenge for the commonly used gradient-based ML models, such as neural networks, as the variance of predictions will be high where the training data is sparse [22, 23]. Second, the distribution of data entries is highly skewed within the table. As can be seen from figure 1, while there is a decent amount of data entries in certain regions, the table becomes much more sparse or even completely empty in other regions. The imbalanced data distribution and the high data sparsity in certain regions make it challenging to train data-intensive ML models (e.g. deep neural networks or DNNs). Last, most advanced ML models, such as DNNs, leverage deeply connected layers to perform non-linear transformations of the input to generate the output. While these models can usually produce highly accurate predictions to match the desired outputs, they are not sufficient for effective application in real-world physics-related applications. Due to their black-box nature and limited data supervision, the learned function may not necessarily follow the physical rules. Therefore, there is a risk that these models may offer false explanations that violate some fundamental physics relations. In this work, we intend to first improve the predictions of the model, then verify if the model can minimize the violation of physics relations.

**Overview of our approach.** To address the key challenges as outlined above, we propose a meta-learning-based Deep Regression model to jointly predict Energy and Pressure (referred to as DREP) after being trained from a finite-sized iFPEOS, aiming to realize a universal EOS model. More specifically, to deal with the imbalanced data distribution and extremely sparse regions, we apply log transformations on both the inputs and outputs before training the DREP model. We compare different types of ML models, showing that most ML models have improved performance with the transformation. NN models are especially good at re-creating the original space and can fit the targets well in all regions. Furthermore, we propose to leverage a meta-learning-based training process to first learn a model that can fit the target properties  $E$  and  $P$  well locally and then generalize to the entire region by giving the model more context



points. We utilize the model's ability to learn from multiple tasks instead of simply running through random batches. This design also provides more flexibility during the test phase when the model is used in practice after being trained.

To achieve the meta-learning-style training process, DREP augments a Neural Process that simulates a stochastic process [24], like a Gaussian Process (GP). On the one hand, DREP inherits the strong function-fitting capacity of a deep neural network to provide accurate predictions. On the other hand, DREP has the advantage of enabling accurate uncertainty modeling. For a standard NN, we can expand the outputs to predict not only  $P$  and  $E$  but also their corresponding variances. However, there is no means to ensure the quality of the predicted variance for a standard NN. By simulating the statistical consistency of a stochastic process through kernel regularization, DREP can faithfully report the predictive uncertainty. In the unseen range, the uncertainty will increase accordingly. Therefore, we are able to identify the uncertainty of the model when the predictions are less reliable through uncertainty quantification. Additionally, we use an ensemble model to improve the stability of model predictions especially when generalizing or applying to downstream tasks. Finally, to verify the thermodynamical consistency of the proposed DREP model, we perform a check of Maxwell's relation regarding energy and pressure. The check is to evaluate the commutativity of the partial derivatives of the predicted pressure and internal energy with respect to density and temperature, relating the two separate outputs. We use auto-differentiation to compute the partial derivatives and find the resulting relative difference between the prediction and Maxwell's relation-induced calculation result to be on the order of 10% or less. Unfortunately, due to the lack of reference free energies, entropies, and chemical potentials in conjunction with the current form of the DREP model, we are unable to verify if other relations regarding thermodynamic consistency, such as the Gibbs-Duhem relation, hold.

**Summary of contributions.** We summarize our contributions as follows:

- We show superior prediction results using an NN model and the remaining challenges in terms of stability, consistency, and uncertainty quantification, which justify the unique design of the proposed DREP model.
- We propose the DREP model, which can more accurately predict pressure and internal energy values at arbitrary density and temperature points.
- We propose kernel regularization to ensure faithful uncertainty quantification.
- We propose to use an ensemble model to improve the stability of model predictions.
- We evaluate the consistency of the DREP model using auto differentiation and a PDE-informed thermodynamic violation measure.
- We provide detailed experimental results to demonstrate the prediction performance of DREP as well as ablation studies and effectiveness on other physics-related downstream tasks.

## 2. Related work

In recent years, machine learning, especially deep neural networks (DNNs) [25], has been widely used not only in computing and data-mining fields [26] but also in many cutting-edge interdisciplinary scientific

research fields [27–29]. With this trend, physics-informed machine learning [30–35] or machine learning for physics [36–38] has attracted increasing attention with promising results. Some recent works aim to develop machine learning (ML) models consistent with real-world physical phenomena. To this end, one popular line of research is to develop data-driven models that rely on observed data (that reflects the underlying mathematical principles) to encode physical rules in the model. Alternatively, specialized neural network architectures with different types of inductive biases [39] (e.g. convolutional networks [25] to ensure translational symmetry) have also been developed to encode the prior physics knowledge in the ML models [34, 40]. DNNs can also be trained by incorporating the underlying physics into loss functions or regularization terms. Finally, hybrid approaches that aim to integrate different physics-informed neural network approaches are also being developed [41–43].

Regression models provide a powerful vehicle to replace expensive numerical calculations or time-consuming simulations by quickly predicting desired physics properties once being trained. Many classic models like linear/polynomial regression and DNNs have been commonly used. However, these models primarily focus on learning from data with limited flexibility to incorporate domain knowledge. When physics-informed knowledge or constraints are considered, kernel-based methods like kernel regression [44, 45], numerical GPs [46, 47], and deep kernel learning [43] have been frequently leveraged with promising results.

Modeling the EOS table is an important task in high-density-energy computational physics. Having a reliable EOS table that covers a wide range is often difficult because of the shortcomings of various computational models [11, 12] and the discrepancies among these methods in the overlapping range [7–9]. iFPEOS provides a more accurate model that covers a wide range of density and temperature values [7]. However, the high computational cost makes it challenging to leverage iFPEOS to generate the desired physics quantities at arbitrary points. While it is possible to train existing regression models, including kernel methods, from limited iFPEOS data points, these approaches fall short in addressing the key challenges as identified earlier in the paper.

When modeling physics-informed problems, differential equations provide an effective means to encode important knowledge or constraints. Although using neural networks to model ordinary or partial differential equations has been studied for a long time [48] and improved recently by PINN (physics-informed neural networks) [30], to our knowledge, there is no prior work that uses indirect PDEs of multiple-outputs (e.g. pressure and internal energy) to verify the thermodynamic consistency without modeling the underlying quantity (e.g. free energy), which is achieved by the proposed DREP model.

### 3. Methodology

In this section, we describe the detailed design of DREP. We start with a formal problem formulation of multi-output regression. We then summarize preliminaries that cover a set of classical regression models. Next, we present the model architecture and discuss how we design the model to achieve each of its key properties in order to address the important EOS modeling challenges.

*Problem formulation.* The main problem—EOS modeling—is a multi-output regression problem. In a standard multi-output regression setting, we have a set of input features  $\mathbf{x} \in \mathbb{R}^D$ , and train the model to predict some continuous output response  $y \in \mathbb{R}$  or multiple responses  $\mathbf{y} \in \mathbb{R}^L$ . In the EOS modeling problem, the input features include density  $\rho$  and temperature  $T$  and the target outputs are pressure  $P$  and internal energy  $E$ :  $\mathbf{x} = (\rho, T)^\top$ ,  $\mathbf{y} = (P, E)^\top$ .

#### 3.1. Preliminaries

A key challenge with the EOS table data is the wide ranges that the physics quantities span. We perform log transformations to make the inputs more accessible for the ML models. The transformation is simple and also makes visualizations of the predictions more accessible. The physics quantities should also satisfy certain boundary conditions (e.g. near density  $\rho = 0$  and temperature  $T = 0$ ). We use extrapolation to generate synthetic training data near the boundary and add these points to the training process. Later we will show how these transformations change the fitting results and the generalizing ability of the model. Below, we give an overview of standard regression models that can be used for EOS modeling.

*Linear/polynomial regression.* Linear regression uses a linear function of the input features  $\hat{y} = \mathbf{w}^\top \mathbf{x}$  to fit the response  $y$ , where  $\mathbf{x} = (1, \rho, T)^\top$  is the feature vector in the EOS problem and  $\mathbf{w}$  is a set of coefficients. However, the expressiveness of a linear function is limited. We can use polynomial feature expansion to

improve the flexibility of the model, which includes the  $p$ th power of each component in  $\mathbf{x}$  with the interaction terms. The coefficients can be learned by minimizing a mean squared error:

$$\mathcal{L}_{PR}(\{\mathbf{x}_n, y_n\}_{n=1}^N) = \frac{1}{N} \sum_{n=1}^N (y_n - \mathbf{w}^\top \mathbf{x}_n)^2. \quad (1)$$

Extension to multiple outputs is straightforward, where the coefficient vector  $\mathbf{w}$  is replaced with a coefficient matrix  $\mathbf{W} \in \mathbb{R}^{D \times L}$  to fit multiple responses  $\mathbf{y} \in \mathbb{R}^L$ :  $\hat{\mathbf{y}} = \mathbf{W}^\top \mathbf{x}$ . Polynomial regression (PR) using high-order polynomial features is prone to overfitting. We can use regularization to reduce overfitting, which we will introduce next together with the kernel trick.

*Kernel methods.* Ridge regression (RR) adds the  $l_2$  norm of the coefficients to address overfitting. The regularized loss function can be formalized as:

$$\mathcal{L}_{RR} = \sum_{l=1}^L \gamma_l \left( \sum_{n=1}^N l_{SE}(\{\mathbf{x}, y_n^{(l)}\}) + \lambda \mathbf{w}^{(l)\top} \mathbf{w}^{(l)} \right) \quad (2)$$

where  $\gamma_l$  is the weight for the  $l$ th output and  $\lambda$  is the regularization weight. For one entry of the output  $y^{(l)}$ , we still use the mean squared error:  $l_{SE}(\{\mathbf{x}, y_n^{(l)}\}) = (y_n^{(l)} - \mathbf{w}^{(l)\top} \mathbf{x}_n)^2$ . We can adopt the matrix view of the RR problem, and the solution can be formed as  $\mathbf{W} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}$ , where  $\mathbf{X}$  is the design matrix (stacking  $\mathbf{x}_n^\top$ ). The term  $\mathbf{X} \mathbf{X}^\top$  is called the *Gram matrix* [49]. We can also allow implicit feature representation using the kernel trick. Kernel ridge regression (KRR) introduces a kernel-represented Gram matrix  $\mathbf{K}$ , where  $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$  with  $k(\cdot, \cdot)$  being a kernel function [49].

GPs also use the kernel trick to build random processes. In a standard GP formulation, the prior is given as a 0-mean Gaussian distribution with the Gram matrix  $\mathbf{K}$  being the covariance matrix  $p(\mathbf{z}) = \mathcal{N}(\mathbf{z} | \mathbf{0}, \mathbf{K})$ . The conditional probability distribution of the target output is also a Gaussian  $p(\mathbf{y} | \mathbf{z}) = \mathcal{N}(\mathbf{y} | \mathbf{z}, \beta^{-1} \mathbf{I}_N)$ . Thus, the marginal distribution of the outputs is still Gaussian  $p(\mathbf{y}) = \int p(\mathbf{y} | \mathbf{z}) p(\mathbf{z}) d\mathbf{z} = \mathcal{N}(\mathbf{y} | \mathbf{0}, \mathbf{C})$ , where  $\mathbf{C}$  is the covariance matrix. The elements of  $\mathbf{C}$  are given by  $C(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j) + \beta^{-1} \delta_{ij}$ . The prediction for an unseen input  $\mathbf{x}_{N+1}$  can also be expressed in the Gaussian form  $\mathcal{N}(y_{N+1} | \mathbf{y})$  where the mean and variance are  $m(\mathbf{x}_{N+1}) = \mathbf{k}^\top \mathbf{C}^{-1} \mathbf{y}$ ,  $\sigma^2(\mathbf{x}_{N+1}) = c - \mathbf{k}^\top \mathbf{C}^{-1} \mathbf{k}$  with  $c = k(\mathbf{x}_{N+1}, \mathbf{x}_{N+1}) + \beta^{-1}$ . The advantage of the GP model is that it can fit the data well locally and provides a natural statistical interpretation that directly gives us a covariance matrix instrumental to quantify the uncertainty.

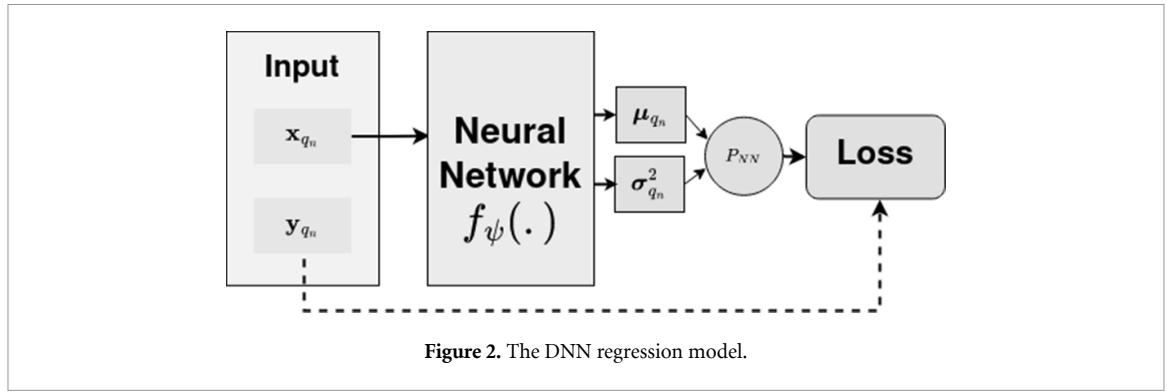
*Deep neural networks (DNNs).* A DNN consists of multiple (usually deeply connected) layers of nodes that play the role of artificial ‘neurons’. All these weights are the parameters of the DNN that are updated during training such that the DNN can approximate the true underlying function and generalize well during inference.

Comparing DNNs with the kernel-based models, we will see the following core differences: First, DNNs can be trained using stochastic gradient descent which is much more scalable than a GP with respect to the number of data points. Second, instead of relying on either the original features or a fixed kernel function, DNNs can learn a latent feature space optimized for the downstream tasks (e.g. regression or classification). In this task, we will take advantage of the second property and utilize the flexibility of DNNs to design a specific model that performs well for the problem. The entire DNN can be expressed as a function:  $\mathbf{y} = f_\Theta(\mathbf{x})$ , where  $\Theta$  denotes all the weights in the network. Besides modeling the outputs  $\mathbf{y} = (P, E)^\top$ , we can make the DNN generate a probabilistic output by outputting both the mean and the variance in the need of quantifying the uncertainty of the prediction. The model is trained by using a log-likelihood-based objective that simplifies to Mean Squared Error loss when the output variance is treated as constant. Such a DNN-based regression model is shown in figure 2.

One major limitation of standard DNNs is that they require a lot of labeled data to be properly trained [50], which makes it challenging to apply for the EOS problem due to highly sparse training data. Furthermore, a DNN trained over limited data may suffer from under-fitting, overfitting [51], or both at the same time. A principled uncertainty quantification mechanism is needed to detect when the model may provide wrong predictions to inform decision-making. The proposed DREP model is designed to address these limitations, which will be detailed next.

### 3.2. Deep regression to jointly predict energy and pressure

We first introduce a task-based meta-learning style training process for the EOS regression problem. Afterward, we develop the DREP architecture for the training process. We then carry out a theoretical analysis that shows the advantage of this unique design over a standard DNN, in terms of sequential inference, thermodynamic consistency, and accurate uncertainty quantification capabilities.



### 3.2.1. A task-based meta-learning style training process

In real-world physics regression problems, we are likely to have a limited number of available training data points (e.g. limited data for the EOS table fitting problem due to expensive Molecular Dynamics simulations). For this work, we assume that we have a limited-data dataset  $\mathcal{D}$  with  $N_{tr}$  labeled data points in the training set ( $\mathcal{D}^{tr} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N_{tr}}$ ), and  $N_{ts}$  data points in the test set ( $\mathcal{D}^{ts} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N_{ts}}$ ).

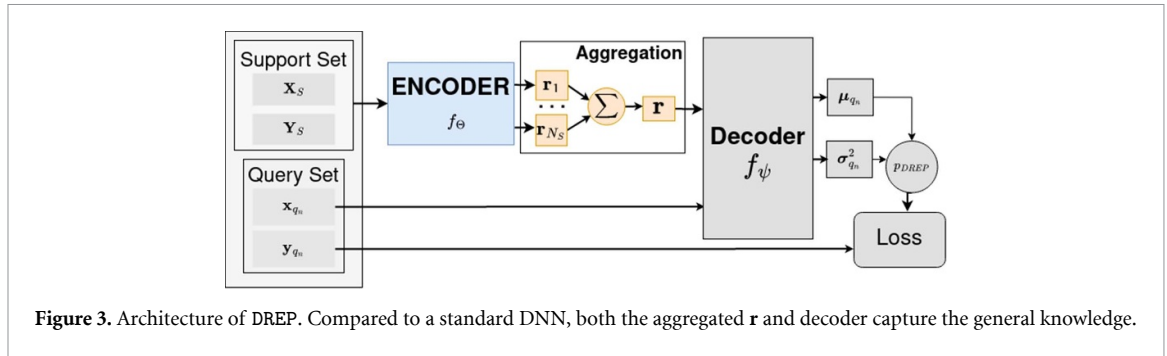
To better utilize the limited training data, inspired by few-shot learning approaches [24], we propose to use a task-based meta-learning-style training process. To this end, we consider two phases: a meta-training phase to acquire the global knowledge of the true underlying regression function, and a meta-testing phase to use the global knowledge in EOS table prediction.

The DREP model accesses the information of  $\mathcal{D}^{tr}$  in the meta-training phase to acquire the global knowledge for the EOS regression task. Specifically, in the meta-training phase, we consider a large number of randomly sampled tasks to acquire the required global knowledge for accurate regression. Each meta-training task consists of a support set  $\mathcal{S} = (\mathbf{X}_S, \mathbf{Y}_S)$  and a query set  $\mathcal{Q} = (\mathbf{X}_Q, \mathbf{Y}_Q)$ . These sets are constructed by randomly sampling  $N_S + N_Q$  data points from the  $N_{tr}$  training data points, assigning the  $N_S$  data points to the support set, i.e.  $\mathcal{S}_n = (\mathbf{X}_{S_n}, \mathbf{Y}_{S_n}) = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{N_S}, (\mathbf{x}_n, \mathbf{y}_n) \in \mathcal{D}^{tr}$ , and assigning the  $N_Q$  points to the query set i.e.  $\mathcal{Q}_n = (\mathbf{X}_{Q_n}, \mathbf{Y}_{Q_n}) = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{N_Q}, (\mathbf{x}_n, \mathbf{y}_n) \in \mathcal{D}^{tr}$ . The support set and the query set of the training task represent two local views of the true regression function, and the DREP model training is formulated such that given one local view of the true function, i.e. the support set view, the model has to accurately predict the query set. In other words, given the support set information, the model has to be able to predict the query set (additional details of model training are provided in section 3.3.1). Such *task-based* local view formulation of the objective enables the model to train on a large number of tasks, with multiple local views, and is expected to guide the model to gain global knowledge of the target function.

From the meta-training phase, the DREP model is expected to acquire the desired meta-knowledge required for accurate downstream EOS regression. We then introduce the meta-testing phase to evaluate the DREP model's acquired knowledge. The meta-testing consists of one test task  $\mathcal{T}_{test}$  in which all the training data constitutes the support set i.e.  $\mathcal{S}_{test} = \mathcal{D}^{tr}$ , and all the test data points constitute the query set i.e.  $\mathcal{Q}_{ts} = \mathcal{D}^{test}, \mathcal{T}_{test} = (\mathcal{S}_{test}, \mathcal{Q}_{test})$ . We consider all the training in the support of the meta-test task, i.e.  $\mathcal{S}_{test} = \mathcal{D}^{tr}$  to ensure the DREP model has a global view of the true regression function. With the global view of the function, the model makes predictions on the test set  $\{\mathbf{x}_i\}_{i=1}^{N_{ts}}$ .

As stated above, the rationale behind using the meta-learning style training for the DREP model is that the data from the EOS table is scarce and sparse, and the underlying function is difficult to learn. To elaborate, although  $P$  and  $E$  increase with  $\rho$  and  $T$  in most regions, there are also the plateau region and other refined local trends. If we simply use one model to learn the entire function, we might either underfit and not describe the training data well, or overfit and lose the ability to generalize. By using the meta-learning style training and having the model learn many local functions first through task-based training, we increase the ability to learn the entire  $P$  and  $E$  functions when given the global view in the meta-testing phase.

**DREP architecture.** Inspired by Conditional Neural Processes [24], we develop an encoder-decoder structure for the EOS regression problem as shown in figure 3. The proposed DREP model considers the neural network structure  $f_{\psi}(\cdot)$  as the decoder, and introduces an encoder  $f_{\Theta}(\cdot)$  that encodes the entire support set information to a vector  $\mathbf{r}$ . The encoder enables the DREP model to capture the knowledge of the support set as a reference for the decoder so that the decoder can consider this reference information to make accurate predictions. Specifically, the reference vector  $\mathbf{r}$  is concatenated with a query point in  $\mathbf{X}_Q$  and passed to the decoder to predict the output.



*Remarks.* We propose to use the DREP model with the encoder-decoder structure because it surpasses the standard DNN by incorporating general knowledge. We further note that the DREP model can realize the DNN as a special instance when it completely ignores the information carried through the encoder structure. This can be proven by a straightforward example. Consider a  $K$  layer DNN represented by  $f_\psi(\cdot)$  with  $m$  dimensional input. Assume the input layer consists of  $D$  neurons. Let  $W^{m \times D}$  represent the weight matrix corresponding to these  $D$  neurons. Consider an equivalent DREP model with  $l$  dimensional encoder representation  $\mathbf{r} \in \mathbb{R}^l$  and  $K$  layers in the decoder structure similar to the DNN. Now, for an equivalent DREP model, consider  $D$  neurons in the first input layer of the decoder. Let  $W^{l+m \times D}$  represent the weight matrix corresponding to these  $D$  neurons. For the DREP model, when  $W^{l \times D}$  (i.e. the components in the weight matrix corresponding to the representation  $\mathbf{r}$ ) are all zero, the representation  $\mathbf{r}$  is ignored, and the DREP model reduces to a DNN model. Equivalently, when the representation  $\mathbf{r}$  is all zero, the representation carries no information, and the DREP model again reduces to the DNN model. In both of these cases, both the network training and inference for the two models are identical. In all other cases, DREP model also considers the information in the representation  $\mathbf{r}$  due to which it is expected to perform better than the neural network as  $\mathbf{r}$  can provide useful reference information for training and inference.

### 3.2.2. Sequential inference and fast adaptation to new training data

The proposed DREP model introduces the encoder-decoder structure that enables the model to capture the knowledge in the training data in two ways: (1) through the parameters of the decoder similar to a standard DNN, and (2) through representation  $\mathbf{r}$  generated by the encoder using the support set. Specifically, during inference, the encoder structure aggregates all the training data to an embedding  $\mathbf{r}$  that acts as the compact representation. For training dataset with  $N_S$  data points,  $\mathbf{r}$  is given as

$$\mathbf{r} = \frac{1}{N_S} \sum_{n=1}^{N_S} f_\Theta(\mathbf{x}_{s_n}, \mathbf{y}_{s_n}) = \frac{1}{N_S} \sum_{n=1}^{N_S} \mathbf{r}_n, \quad (\mathbf{x}_{s_n}, \mathbf{y}_{s_n}) \in \mathcal{S}_n. \quad (3)$$

The representation is permutation invariant over the input data [52] that encodes all the training data and aggregates the resultant  $N_S$  embeddings. This representation can be expressed via a sequential update rule:

$$\mathbf{r} = \frac{1}{N_S} \sum_{n=1}^{N_S} \mathbf{r}_n = \frac{1}{N_S} \mathbf{r}_{N_S} + \frac{N_S - 1}{N_S} \frac{1}{N_S - 1} \sum_{n=1}^{N_S - 1} \mathbf{r}_n = \frac{1}{N_S} \mathbf{r}_{N_S} + \frac{N_S - 1}{N_S} \mathbf{r}_{\text{old}} \quad (4)$$

where  $\mathbf{r}_{\text{old}} = \frac{1}{N_S - 1} \sum_{n=1}^{N_S - 1} \mathbf{r}_n$  is the aggregated representation of the first  $N_S - 1$  data points, and  $\frac{1}{N_S} \mathbf{r}_{N_S}$  is the encoder embedding for the new  $N_S^{\text{th}}$  data point. This sequential update rule enables the model to discard the training data once observed, and also incorporate new information/observations for improved prediction during the inference phase. The sequential inference capability of DREP can enable some practical use cases for the EOS problem. For example, sequential inference is useful when we might have different EOS data that can be used as the support set at test time. Usually, when new ground truth data are available, we need to re-train the DNN model, which is time-consuming. However, for the DREP structure, we do not need to re-train the model but only need to include the new ground truth data in the support set that serves as reference data to support prediction. This shows the generalization ability of the model from an ML perspective. Moreover, the sequential update also enables the model to be effective when some regions of density/temperature are not available during model training. To this end, we do not need to store the entire dataset used for training. We can keep the learned representation and use it in future tasks. Since EOS

problems may involve different computational physics models, this functionality can be very useful in practice.

### 3.3. Challenging tasks for NN: uncertainty quantification and downstream tasks

*NN uncertainty quantification.* We use the common practice from [53] to connect the predicted variance of NN to the model outputs,  $P$  and  $E$ . More specifically, we modify the loss function by including the variance to each prediction and a regularization term that includes the variance itself:

$$L_{var} = \frac{1}{N} \sum_{i=1}^N \frac{1}{2} \left( \frac{\|P_{NN} - P\|^2}{\sigma_P^2} + \log \sigma_P^2 + \frac{\|E_{NN} - E\|^2}{\sigma_E^2} + \log \sigma_E^2 \right) \quad (5)$$

where  $P$  and  $E$  are equally balanced. However, as shown in the later section (figure 10), the results are usually not meaningful as NN does not consider the relationship to the training data here.

#### 3.3.1. DREP Uncertainty quantification results

When the training data is limited, it is desirable that the model remains uncertain on its predictions in regions far away from the observed data samples. To this end, we introduce a novel variance regularization term that aims to guide the model to be uncertain in regions with limited or no training data samples:

$$\mathcal{L}_{KER} = \frac{1}{N_Q} \sum_{n=1}^{N_Q} \frac{1}{\sigma_{\mathbf{x}_{q_n}}} \text{Dist}(\mathbf{x}_{q_n}, \mathcal{S}_n) \quad \mathbf{x}_{q_n} \in \mathbf{X}_{Q_n} \quad (6)$$

where  $\sigma_{\mathbf{x}_{q_n}}$  represents the predicted variance for the query input  $\mathbf{x}_{q_n}$ , and  $\text{Dist}(\mathbf{x}_{q_n}, \mathcal{S})$  represents the distance between the query point  $\mathbf{x}_{q_n}$  and the point in the support set  $\mathcal{S}_n$  nearest to the query point  $\mathbf{x}_{q_n}$ . To minimize this loss, the variance should be high for (1) data points far away from the observed data, and (2) in regions of missing data. DREP model enables us to introduce such novel regularization to accurately guide the uncertainty. In the regions near to observed data, the distance will be low leading to overall low loss.

We train the model to maximize the conditional log-likelihood

$$\mathcal{L}_{DREP} = \sum_{n=1}^{N_Q} -\log(p_{DREP}(\mathbf{y}_{q_n} | \mathcal{S}_n, \mathbf{x}_{q_n})) = \sum_{n=1}^{N_Q} -\log(\mathcal{N}(\mathbf{y}_{q_n} | \mu_{q_n}, \sigma_{q_n}^2)) \quad (7)$$

where  $\mathcal{S}_n$  is the support set of the training task  $T_n$ ,  $\mathbf{y}_{q_n}$  represents the query set output for query set input  $\mathbf{x}_{q_n}$ ,  $(\mathbf{x}_{q_n}, \mathbf{y}_{q_n}) \in \mathcal{Q}_n$ ,  $\mathcal{N}(\cdot)$  represents the gaussian distribution,  $\mu_{q_n}$  represents the predicted mean,  $\sigma_{q_n}^2$  represents the predicted variance, and  $p_{DREP}$  represents the DREP model. In addition, we introduce the kernel-based regularization term  $\mathcal{L}_{KER}$  (equation (6)). The overall loss of the model is given by

$$\mathcal{L} = \mathcal{L}_{DREP} + \lambda_1 \mathcal{L}_{KER} \quad (8)$$

where  $\lambda_1$  is the regularization coefficient that controls the impact of variance regularization on the overall model training.

#### 3.3.2. Prediction/thermodynamics consistency checking

Using the model designs and regularization methods from previous sections, we have established a multi-output model that can fit the EOS data well despite the range and sparsity issues. However, DREP simulates stochastic processes and still produces some variance when there are no nearby reference data points. This would create some wiggles in the predicted  $P$  and  $E$  curves. Additionally, in the large-density regions, the model predictions might have some larger absolute errors due to overfitting. To address this issue, we can train multiple models with randomized initialization and meta-learning-style tasks. The ensemble of these randomized models creates more reliable prediction results. Next, we also verify that by improving the prediction stability of the model, we also improve the consistency when applied to downstream tasks.

*Thermodynamic consistency from gradient-based PDE measure.* The proposed DREP model can fit the pressure and internal energy well with limited training data. However, to safely utilize these prediction results, we would need to verify them in terms of thermodynamic consistency. We know that both pressure and internal energy can be derived from the Helmholtz free energy  $F$ . This can be combined with the fundamental thermodynamic relation:  $dU = -SdT - PdV$ . Thus, we have

$$P = -\left. \frac{\partial F}{\partial V} \right|_T, \quad E = F - T \left. \frac{\partial F}{\partial T} \right|_V. \quad (9)$$

where  $V$  is the volume. We perform partial differentiation of  $P$  and  $E$ , then using the chain rule we have:

$$\frac{\partial P}{\partial T} = -\frac{\partial^2 F}{\partial T \partial V}, \quad \frac{\partial E}{\partial V} = \frac{\partial F}{\partial V} - T \frac{\partial^2 F}{\partial V \partial T} = -P + T \frac{\partial P}{\partial T} \quad (10)$$

$$P = T \frac{\partial P}{\partial T} - \frac{\partial E}{\partial V} \quad (11)$$

with  $\frac{\partial V}{\partial \rho} = -\frac{m}{\rho^2}$ , where  $m$  is the mass corresponding to  $V$ , we get

$$P_{\text{CONSISTENCY}} = P = T \frac{\partial P}{\partial T} + \frac{\rho^2}{m} \frac{\partial E}{\partial \rho} \quad (12)$$

which we will use as the consistency criterion.

We propose to use the computed gradients from the DREP model to generate a  $P_{\text{CONSISTENCY}}$  term and compare it with the ground truth  $P$  or the DREP prediction  $P_{\text{pred}}$  if the ground truth is not available. If the difference is small, we can conclude that the  $P$  and  $E$  predictions are consistent with each other. Together with the accuracy of the actual  $E$  predictions, we can conclude that the model makes thermodynamically consistent predictions.

## 4. Evaluation results

We first introduce the iFPEOS dataset and experiment details in section 4.1. We then present the quantitative results that show the superiority of our proposed model over the baseline regression models in section 4.2. In section 4.2, we compare the prediction results using averaged MRE results from training-validation-splits. Afterward, we present the consistency and uncertainty quantification results of our proposed DREP model. The final results are from the proposed model trained with all available training data. Finally, we carry out multiple ablation studies to investigate the contribution of different components of the proposed model.

### 4.1. Dataset description

The iFPEOS dataset is visualized in figure 4. It consists of 1637 data points:  $\mathcal{D} = \{(\rho_i, T_i, E_i, P_i)\}_{i=1}^{1637}$ , where 1228 are observations from experiments that reflect samples from the true underlying function  $f$  such that  $(E, P) = f(\rho, T)$ , 63 are interpolated on the isochores  $0.0196\text{--}0.0841 \text{ g cm}^{-3}$  for temperature points  $0.086\text{--}10.77 \text{ eV}$ , and all remaining ones are extrapolations [54] from the observed data points. We consider extrapolations at density  $1 \times 10^{-5} \text{ g cm}^{-3}$  and temperature  $8.62 \times 10^{-6} \text{ eV}$  as boundary data points.

As mentioned in section 3.2, to address the wide range and high sparsity of the given dataset, we apply log transformations. The input scale is already presented in figure 4. We also visualize the outputs in figure 5. We consider all the interpolations and extrapolations as part of the training data. From them, we randomly select 80% for training:  $\mathcal{D}^{\text{tr}} = \{(\rho_i, T_i, E_i, P_i)\}_{i=1}^{1391}$ , and the rest for testing:  $\mathcal{D}^{\text{ts}} = \{(\rho_i, T_i, E_i, P_i)\}_{i=1}^{246}$ . We repeat the random train-test split 5 times and present the average test set results across the 5 runs.

### 4.2. Prediction results and comparisons

In this section, we present experimental results that: (1) compare the prediction results of the proposed DREP model with several commonly used baseline regression models; (2) verify thermodynamic consistency of the ML-EOS predictions; and (3) demonstrate the stability and generalization abilities of the proposed model from both ML and physics perspectives.

We use the mean relative error (MRE) as the main metric to show the overall performance, which is defined as:

$$\text{MRE} = \frac{1}{N} \sum_{n=1}^N \frac{|\hat{y}_n - y_n|}{|y_n|} \quad (13)$$

where the target response can be either  $P$  or  $E$ . We first summarize the overall MRE comparison in table 1. As can be seen, the proposed DREP outperforms all the baseline models on both energy and pressure predictions. We will demonstrate more detailed results including  $P/E - T$  curves at different density points and show how the baseline models suffer from the main challenges as summarized earlier in the paper.

*Prediction results of DREP.* In this section, we present a detailed visualization of the EOS fitting results by the DREP model. The overall objective of the model is to take density  $\rho$  and temperature  $T$  as inputs and make predictions on pressure  $P$  and internal energy  $E$  as outputs. In the training stage, we make 80% of the entire table available and iteratively generate tasks from it. Each task consists of 50 context points and 50 target points. We consider the trained DREP model and analyze the model's regression capability for different

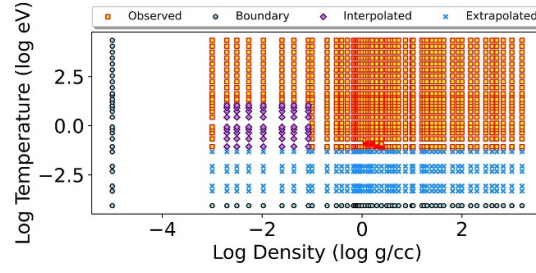


Figure 4. Dataset composition visualization.

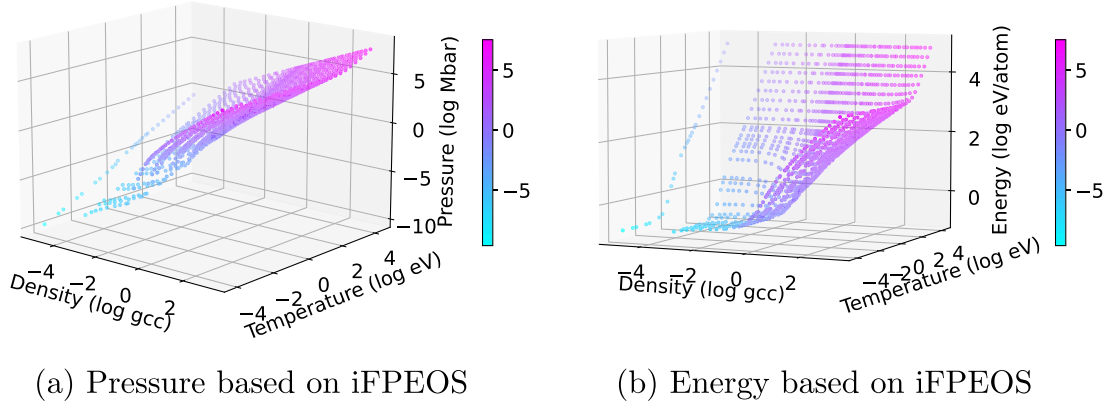


Figure 5. Visualization of the dataset in the log scale .

Table 1. Comparison results.

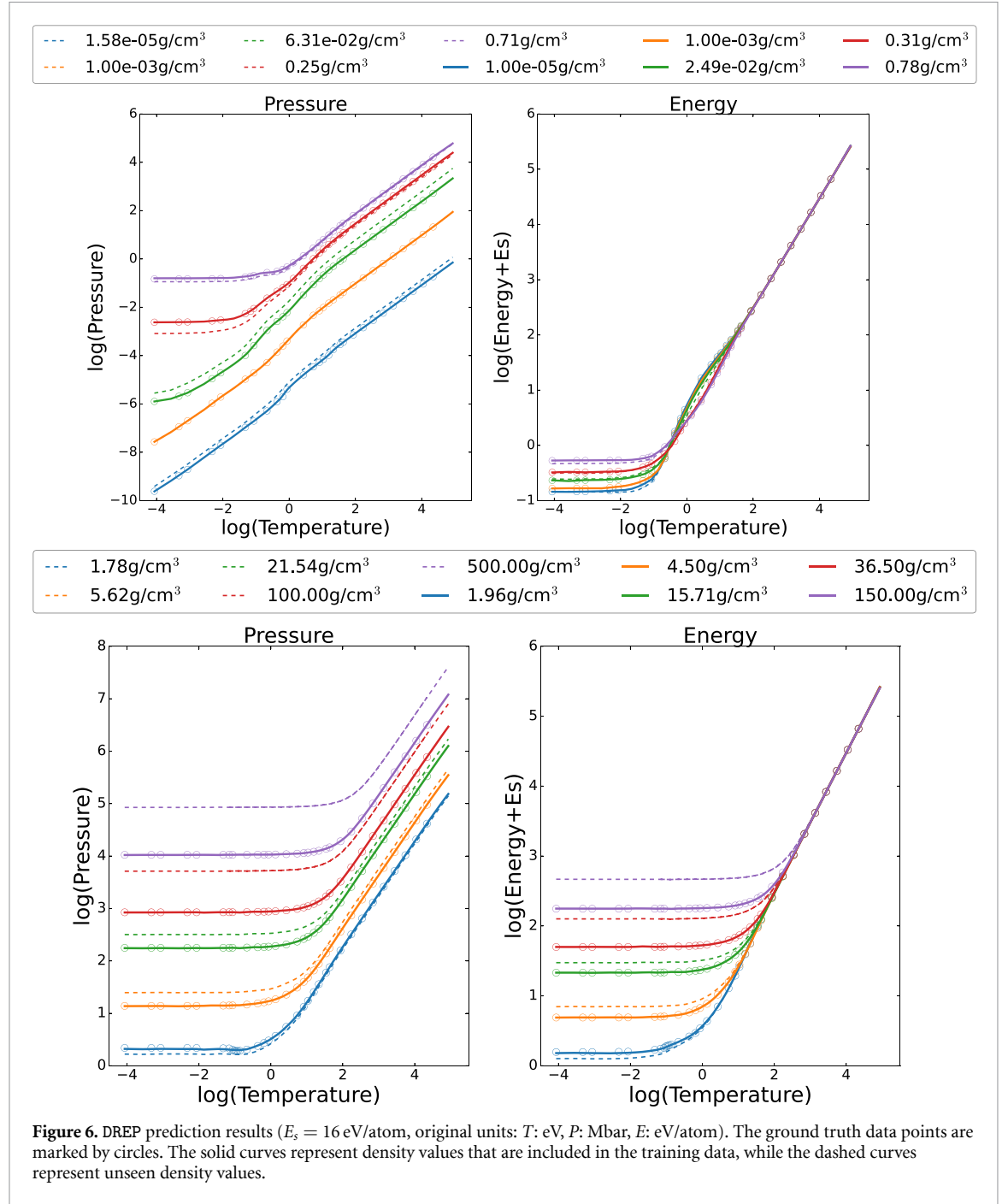
Model	Average MRE P (%)	Average MRE E(%)
RR	397	492
RR <sub>poly=13</sub>	7.11	6.00
KRR	5.84	5.95
GP	24.3	31.9
NN	1.11	1.19
DREP	0.90	1.03

density values using the pressure-temperature and energy-temperature Curves. For each density value, we consider the temperature in the range of  $8.6 \times 10^{-6}$  eV–22 060 eV, and plot the ground truth values along with the energy and pressure predictions. In figure 6, each pressure-temperature curve is generated by predicting on 1000 temperature values for each density value. For better visualizations, we use the log scale. The original units for the quantities are:  $\text{g cm}^{-3}$  for density, eV for temperature, Mbar for pressure, and eV/atom for internal energy. The temperature values are evenly distributed in the *log* space. The reference points (ground truth data) from the original EOS table are shown as circles in the figure. The solid curves represent density values that are included in the training data, while the dashed curves are predictions for density values not present in the training dataset. It is worth to note the unseen density curves are also smooth and show reasonable trends compared to the adjacent curves that include ground truth points.

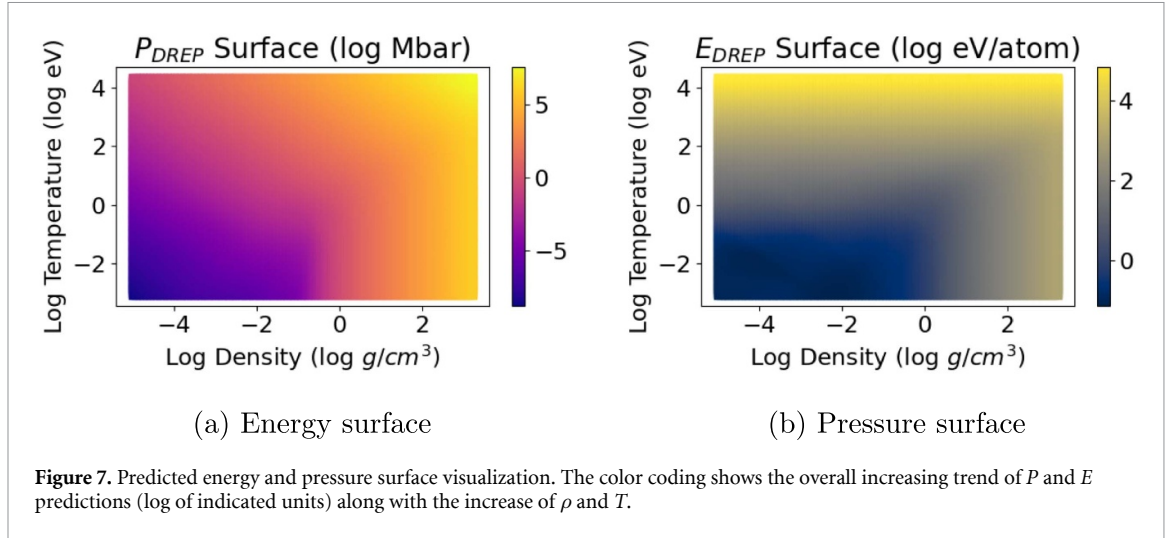
We next visualize the predicted surface of energy and pressure for density in the range  $1 \times 10^{-5} \text{g cm}^{-3}$ – $1597 \text{g cm}^{-3}$  and temperature in the range  $8.6 \times 10^{-4}$  eV–22 060 eV. For the energy trend, we shifted the DREP model prediction by  $E_S = 16$  eV/atom before the log transformation. Figure 7 visualizes the two surfaces predicted by our DREP model. We also visualized the relative error distribution in figure 11(a). The error is mostly evenly distributed over the density-temperature range in which the model is trained in. The relative error can be higher when the ground truth value is small, which is expected.

*Prediction results of baseline models.* For a more thorough comparison, we present some detailed prediction results from representative baseline models, including ridge regression (RR) and GP regression.

For the RR model, the overall MRE is around 444.5%. Apparently, a linear model can not capture the complex (and highly nonlinear) relationship between the outputs and the inputs. One solution is to



construct nonlinear polynomial features based on the inputs. To this end, we have tested polynomial orders of 2, 3, 5, 7, 13, 17, and the corresponding MRE results are:  $L_{\text{MRE}}(\text{Poly}_2) = 72.9\%$ ,  $L_{\text{MRE}}(\text{Poly}_3) = 60.4\%$ ,  $L_{\text{MRE}}(\text{Poly}_5) = 29.7\%$ ,  $L_{\text{MRE}}(\text{Poly}_7) = 16.3\%$ ,  $L_{\text{MRE}}(\text{Poly}_{13}) = 5.2\%$ ,  $L_{\text{MRE}}(\text{Poly}_{17}) = 205.5\%$ . After a polynomial order of 13, the model starts to severely overfit and the MRE quickly increases. The MRE results with polynomial order of 13 look promising. However, by closely checking the pressure-temperature curves of  $\text{Poly}_{13}$ , it shows that the model is highly unstable between ground-truth data points. By using the kernel approach instead of expanding to polynomial inputs, Kernel ridge regression (KRR) can further improve the smoothness of the RR model, achieving an MRE at around 5%. However, as can be seen in figure 8, the curves still have severe wiggles in many density regions. Particularly, the prediction struggles to stay true to the reference data trend in the low temperature region. We can see from figure 5 too that the low temperature region changes drastically from low density to high density, which could be the cause of the poor performance from baseline models in this range. Finally, the GP model exhibits some large predictive variances for regions where training data is completely missing. It is also worth noting that in table 1, we



show the overall MRE after omitting some highly nonphysical predictions (with relative errors being much larger than 100%), and the results are still far from ideal.

*Sequential inference experiments.* The proposed model can integrate knowledge from different sizes of support sets for downstream regression problems during the inference phase as a sequential inference model. Here, we show how the model's sequential inference capability can be useful in the EOS problem. If we use a support set of size 0, the model reduces to a simple NN model. When we change the size of the support set, we change the amount of information we force the model to consider during inference for prediction on the test inputs.

We consider the DREP model trained for 1000 epochs with novel kernel-based regularization strength  $\lambda_1$  of 0.1. As shown in figure 9, when there is only one data point in the support set, the MRE's for both  $P$  and  $E$  are very high. As the support set size increases, the model has more knowledge about the true regression function, and the model obtains a better estimate of the true underlying function which leads to improved performance on downstream regression tasks.

#### 4.3. Uncertainty quantification results

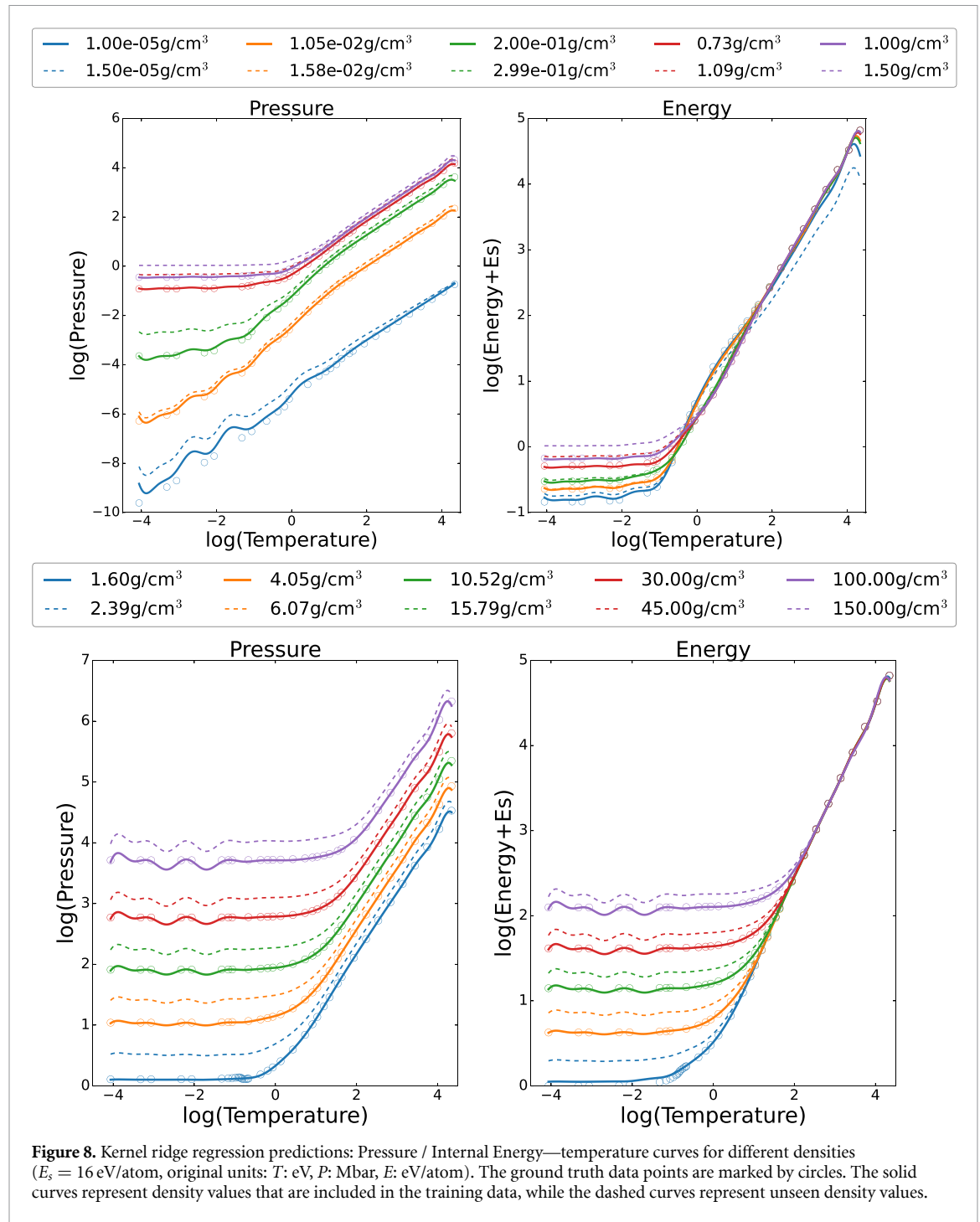
DREP introduces a kernel-based regularizer that enables the model to have accurate uncertainty quantification capabilities, which will be investigated in this section.

*NN uncertainty quantification results.* First, we show the issue with the predicted variances of NN. In figure 10, the variance is not very accurate in that it can not provide an interpretation of how the model is performing outside of the training regime. The predicted variance of  $P$  is larger only in the low temperature region, while the predicted variance of  $E$  is almost uniform across the entire data space. The desired behavior of the predicted variance should indicate whether the model is certain about the predictions. Next, we show that our proposed DREP structure enables us to use a novel regularizer to do exactly that.

*Regularization and uncertainty quantification.* The support set offers information that makes DREP different from a standard NN. In supervised ML, when the target prediction region is very different from the labeled data that we have seen during training, the prediction results are not as reliable as in-range predictions. The model should be able to quantify such unreliability. In our framework, these cases can be captured by the uncertainty through the predicted variance. To ensure that the model predicts a higher variance when the test data point is far away from the available training data, we propose to add a kernel regularization term to the loss function (section 3.3.1). In the EOS problem, we introduce the variance regularization loss during training as:

$$\mathcal{L}_{\text{KER}} = \sum_{n=1}^{N_Q} \text{Dis}(\mathbf{x}_{q_n}, \mathcal{S}_n) \times \left( \frac{1}{\sigma_{P_n}} + \frac{1}{\sigma_{E_n}} \right) \quad (14)$$

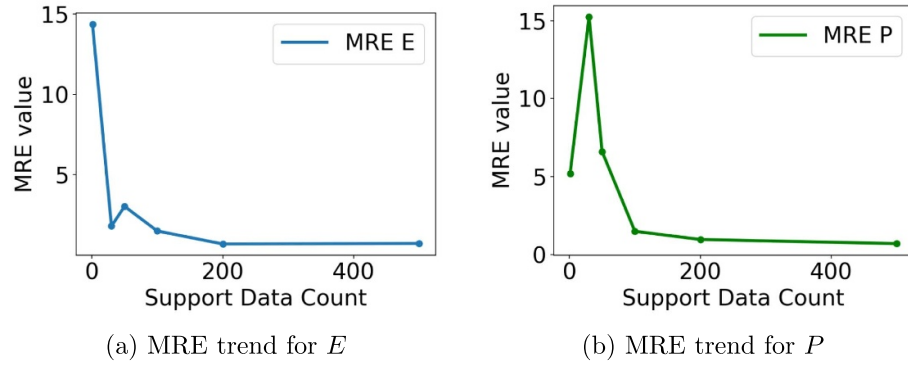
where  $\sigma_{P_n}$  and  $\sigma_{E_n}$  represent the predicted pressure and energy for the query input  $\mathbf{x}_{q_n}$ . We train the DREP model on the 1228 iFPEOS dataset points directly observed from the experiments. After training for 1000 epochs, we plot the predicted variance for missing data regions and data regions far away from the



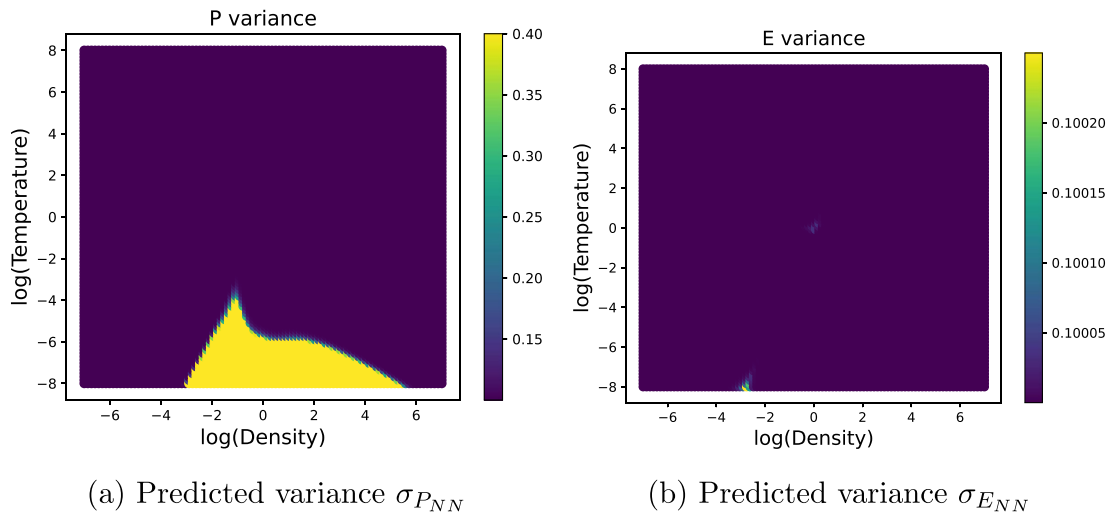
observations in figure 11. We consider the model's prediction and consistency in the density range  $0.0001 \text{ g cm}^{-3}$ – $1597 \text{ g cm}^{-3}$  and temperature range of  $0.068 \text{ eV}$ – $22061 \text{ eV}$ . As can be seen, the model outputs high variance in regions of low density and temperature, which corresponds to the missing data region (see figure 4), where no reference data is available for the model to learn from. Moreover, this high variance also correlates to a high relative error region, a desirable property of an uncertainty-aware model.

*Out-of-distribution detection experiments.* In the above experiments, the input range is still close to the training data.

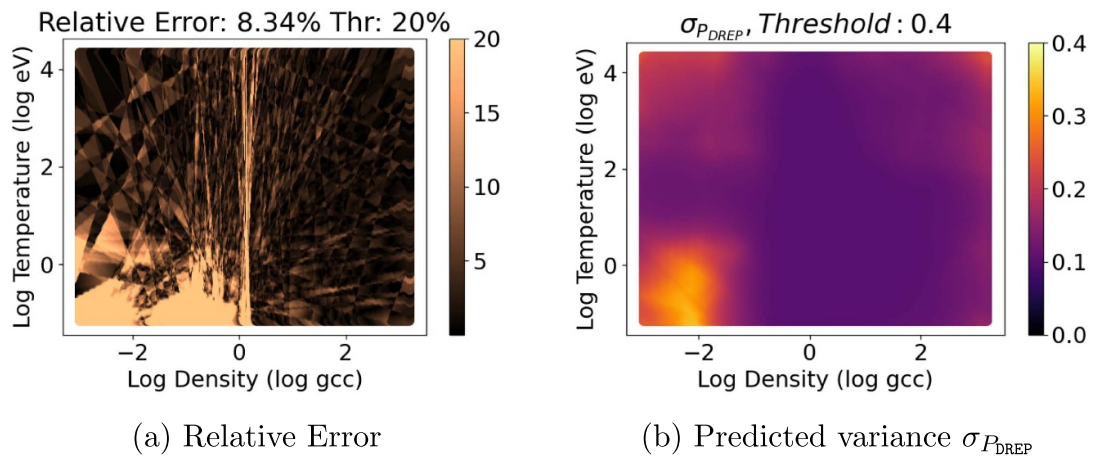
We have shown that our model can generalize better than basic models in these cases. However, if the test inputs are even further away from the known region, all the models are expected to make more unreliable predictions. In this case, we would like the model to output a high uncertainty score that indicates a



**Figure 9.** Trends of MRE vs. Number of data points in the support set. Number of points in the support data set is indicated by x axis values. We can see the MRE decreases as we increase the size of the support set, which indicates that the support set data points provide useful information that could help the model make accurate predictions.



**Figure 10.** Trends of NN predicted variance. The predicted variance for pressure does not match the training data distribution and the predicted variance for internal energy is mostly non-informative.



**Figure 11.** Trends of relative error and variance of DREP with  $\lambda_1 = 0.1$ .

potentially unreliable prediction. Our DREP model is designed with this objective and leads to desired uncertainty behavior. Figure 12 shows the variance output of the DREP model for such out-of-distribution regions, where the model variance increases as the model makes predictions on regions far away from the training data.

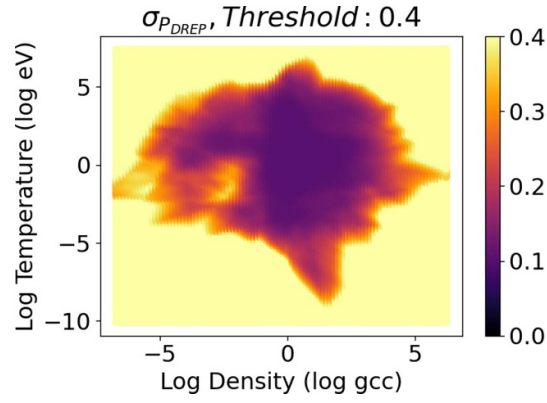
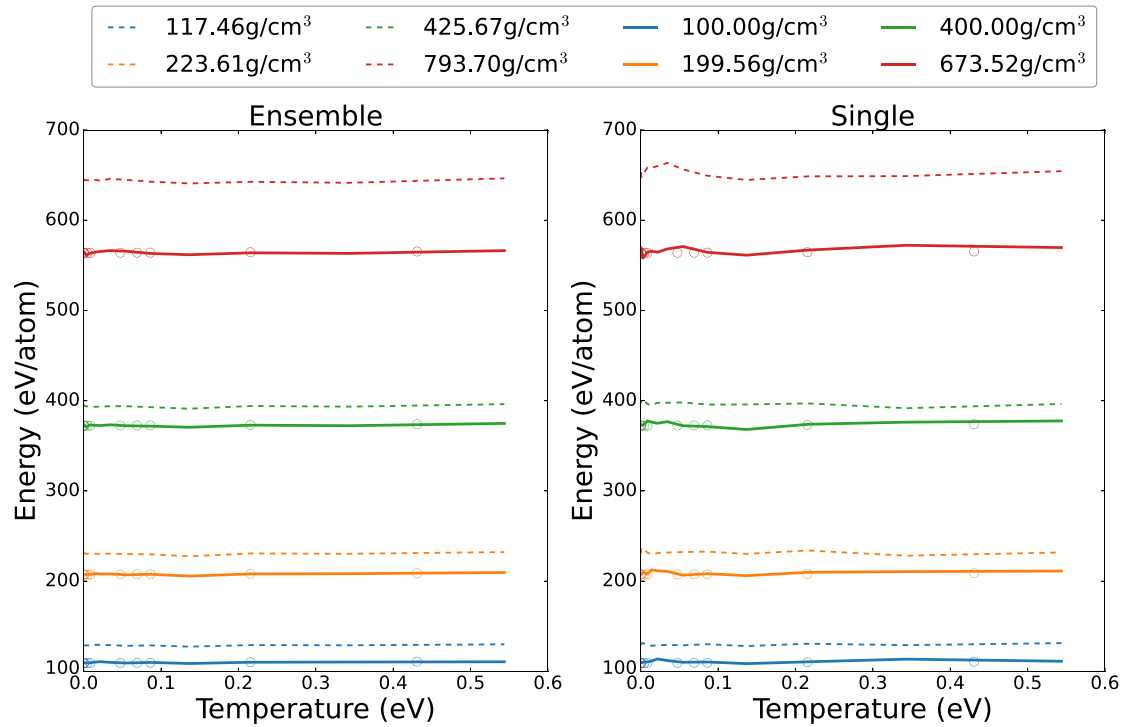


Figure 12. Predicted variance.

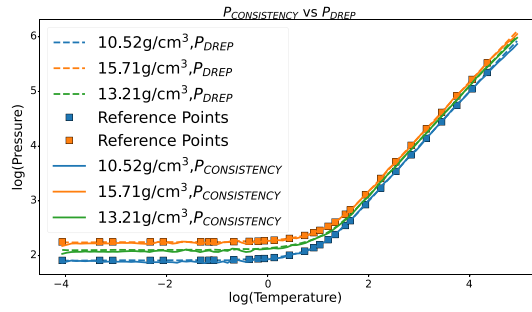


**Figure 13.** DREP prediction results with/without ensembles. The ground truth data points are marked by circles. The solid lines represent density values that are included in the training data, while the dashed lines represent unseen density values. We can see that the single model predictions (right figure) have wiggles in low-temperature regions. The average training MRE of the single models is 0.89% for  $P$  and 1.11% for  $E$ , while the ensemble predictions on the training data has an MRE of 0.77% for  $P$  and 0.51% for  $E$ .

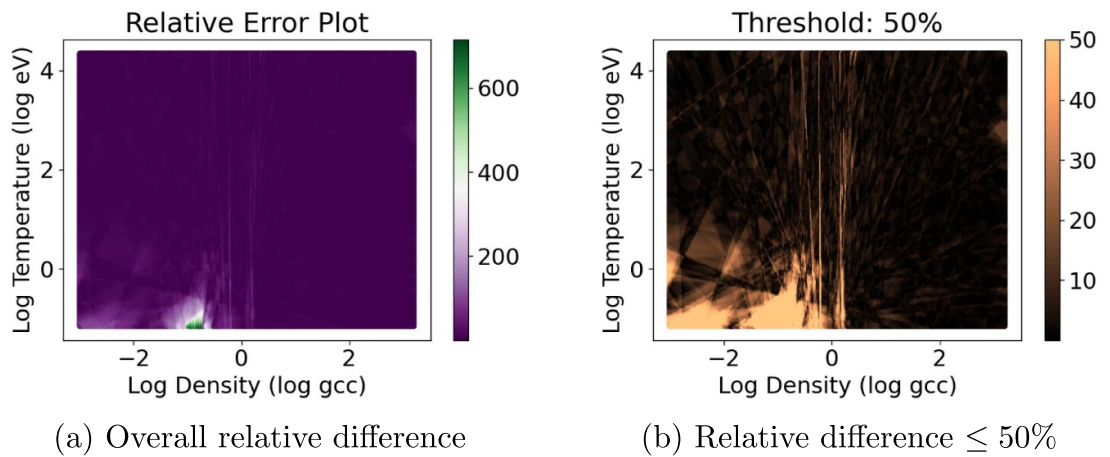
#### 4.4. Consistency analysis

We next carry out experiments to study the physical consistency of the DREP model. We first show the results on using ensembling to improve the model consistency. We then examine the consistency in the model predictions using automatic differentiation [55]. Finally, we plot the Hugoniot plot to further verify the model consistency.

*Consistency results with/without ensembling.* Although the overall MRE is already low with DREP, we can still observe few wiggles or oscillations in smaller-value regions (figure 13). To address this issue, we propose to leverage the ensemble method by training multiple randomly initialized models with different random meta-learning-style tasks. In figure 13, we compare the high-density-low-temperature predictions of a single model and the ensemble model, which shows that the ensemble model improves the prediction consistency to a large extent. This, together with the thermodynamic consistency in the following subsection, will greatly benefit downstream tasks. We will use the Hugoniot as an example at the end of the section to demonstrate the overall effectiveness.



**Figure 14.** Trends of DREP model prediction and consistency-based computation of  $P$ . The relative difference in this density range is 3.03%.



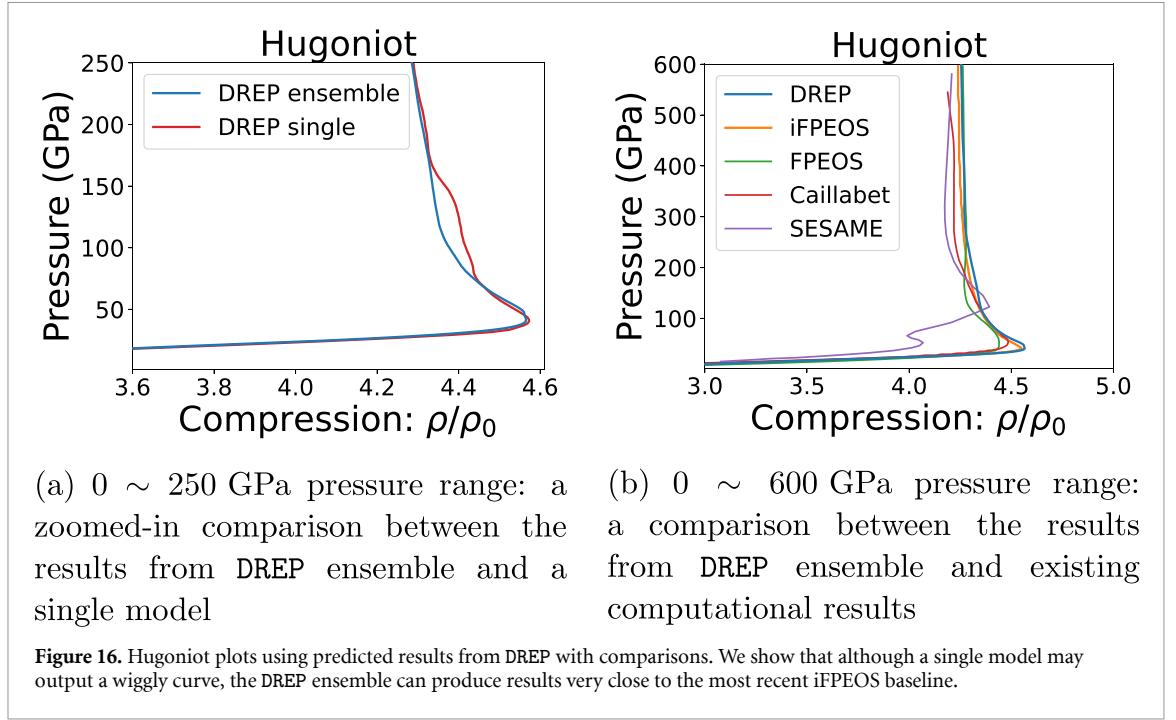
**Figure 15.** Relative difference between predictions  $P_{DREP}$  and  $P_{CONSISTENCY}$ .

*Gradient-based consistency results.* It is important that the EOS predictions are thermodynamically consistent. To evaluate whether desired thermodynamic consistency can be achieved by the proposed model, we compare the model's pressure prediction ( $P_{DREP}$ ) with the pressure value ( $P_{CONSISTENCY}$ ) obtained by solving the consistency criterion in equation (12). In figure 14, we show both  $P_{DREP}$  and  $P_{CONSISTENCY}$  for three different density values at different temperatures. It can be seen that the  $P_{DREP}$  curves match closely with the  $P_{CONSISTENCY}$  curves. In addition, the model prediction  $P_{DREP}$  also matches the ground truth  $P$  points (GT points) almost perfectly. It is also worth noting that no data points are available in training for a density of  $13.00 \text{ g cm}^{-3}$ . The DREP model is still able to output physically reliable pressure values that are bounded between the pressure curves of density  $10.52 \text{ g cm}^{-3}$  and  $15.71 \text{ g cm}^{-3}$ . Moreover, it is well aligned with the corresponding  $P_{CONSISTENCY}$  curve as shown in figure 14.

Next, we show the difference (using MRE) between  $P_{DREP}$  and  $P_{CONSISTENCY}$  for the DREP model trained over all the training data (including interpolated, and extrapolated data) in figure 15. The averaged overall relative difference in the entire density temperature range is around 9%, and the distribution is almost uniform except for the  $\rho \sim 0.1 \text{ g cm}^{-3}$  region where the relative difference reaches a very high value of up to 600% (see figure 15(a)). It can be interesting future work to study this  $\rho \sim 0.1 \text{ g cm}^{-3}$  region to better understand the ML model's inconsistent prediction around this region. In low-density-temperature regions, the relative difference exceeds a threshold of 50% (see figure 15(b)). In all other regions, the model's predictions are mostly consistent, the relative difference is reasonably low, and the model is accurate.

*Fitting the Hugoniot curves.* Besides analyzing the predictions and PDE-induced consistency measures, another important verification of the model consistency is to test it for downstream tasks.

The Hugoniot relations, which provide the thermodynamic conditions in shock-compressed matter, are an important and convenient benchmark for the accuracy of EOS models. The Hugoniot equation relates the



**Table 2.** MRE results without log transformation.

ML model	Average MRE P (%)	Average MRE E (%)
RR	144	27.9
KRR	41.3	24.8
GP	92.3	85.0
NN	253	257

internal energy, pressure and density ( $E_0, P_0, \rho_0$ ) in the unshocked side of the shock front to those in the shocked side ( $E, P, \rho$ ):

$$E - E_0 = \frac{1}{2} (P + P_0) \left( \frac{1}{\rho_0} - \frac{1}{\rho} \right), \quad (15)$$

$E_0$  and  $P_0$  have been obtained for the initial conditions corresponding to those in reported experimental measurements ( $\rho_0 = 0.173 \text{ g cm}^{-3}$ ,  $T = 19 \text{ K}$  [4]) using the methodology presented in existing work [7]. The Hugoniot curves, corresponding to the pressure-compression points ( $P, \rho/\rho_0$ ) which satisfy equation (15), are presented in figure 16, where we show curves generated by: (1) predicted table from a single model; (2) predicted table from ensemble model, and compare the ensemble model results with existing results [7, 12, 15]. As we can see, using a single model leads to a more wiggly curve that has sudden slope changes compared to the ensemble model. If we compare with the most recent existing results [7], we find that these wiggles can not be interpreted as reasonable physical behaviors. The ensemble model shows smooth curves which are close to existing results. Thus, although the overall predictive performance (e.g. using MRE as a metric) of a single model is close to the ensemble model, the latter shows an improved ability to adapt to downstream tasks.

#### 4.5. Ablation studies

We first investigate the impact of log transformation. We then study how the trade-off parameter  $\lambda_1$  and the width  $L$  of neural network layers affect the model's performance.

*Prediction results without preprocessing.* In table 2, we show the prediction results without log transformation. As we can be seen, the model performance without log transformation is much worse than reported in table 1. Specifically, the NN model suffers more than other models because it does not consider the similarity between data points as in kernel-based methods. This shows that both the transformation and the model design are important for a good prediction performance.

**Table 3.** Impact of  $\lambda_1$ .

$\lambda_1$	Average MRE P (%)	Average MRE E (%)
0	0.90	1.03
0.001	0.89	1.09
0.1	0.91	1.25
10	5.80	3.01

**Table 4.** Impact of layer width  $L$ .

$L$	Average MRE P (%)	Average MRE E (%)
32	3.49	4.34
64	2.56	2.63
128	1.61	2.41
512	0.98	1.35
1024	0.90	1.03
2048	1.02	1.32

*Impact of regularization parameter  $\lambda_1$ .* The regularization parameter  $\lambda_1$  controls the contribution of the kernel regularizer. We show the impact on the prediction results in table 3. As can be seen, stronger regularization hurts the generalization performance. With a reasonable regularization value (e.g. 0.001–0.1), the model has accurate uncertainty behavior, and reasonable prediction performance in terms of average MRE of both P and E.

*Impact of width  $L$ .* The proposed DREP model consists of an encoder block and the aggregation module, followed by the decoder block (see figure 3). The encoder and decoder blocks are neural networks with  $L$  neurons in each layer. We evaluate models with different widths  $L$ . All the models are trained for 10000 epochs (each epoch consists of 2000 training tasks) and evaluated on the same test set. Table 4 shows the results. For a model with low  $L$  values, it is likely to under-fit the training data. In contrast, for large  $L$  values, the models tend to overfit to the training data. As can be seen from the table, the best result is achieved by a model with  $L = 1024$ .

## 5. Conclusion

In this work, we conduct deep learning-based regression to jointly predict energy and pressure at an arbitrary point, aiming to facilitate and accelerate the construction of universal equation-Of-State (EOS) models. We introduce log transformations and meta-learning-inspired training that lead to an accurate, thermodynamically consistent, and uncertainty-aware deep regression model. Experiments across multiple baselines and settings demonstrate the effectiveness of the developed model. The designed training mechanism proves to work well under wide-ranged and sparse data settings. The uniquely designed kernel-based regularizer ensures accurate uncertainty quantification even with highly sparse training data. The ensembling technique further improves the prediction consistency of the model, which is also demonstrated in the improved thermodynamic consistency and downstream tasks.

## Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

## Acknowledgments

This material is based upon work supported by the Department of Energy National Nuclear Security Administration under Award No. DE-NA0004144, the University of Rochester, and the New York State Energy Research and Development Authority. Partial funding for this research was provided by the Center for Matter at Atomic Pressures (CMAP), a National Science Foundation (NSF) Physics Frontier Center, under Award PHY-2020249. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation

## ORCID iDs

Dayou Yu  <https://orcid.org/0009-0002-2373-4907>

Deep Shankar Pandey  <https://orcid.org/0009-0006-1404-3716>

Deyan Mihaylov  <https://orcid.org/0000-0002-8874-5503>

Valentin V Karasiev  <https://orcid.org/0000-0003-3445-6797>

S X Hu  <https://orcid.org/0000-0003-2465-3818>

Qi Yu  <https://orcid.org/0000-0002-0426-5407>

## References

- [1] Kritcher A L *et al* 2022 Design of inertial fusion implosions reaching the burning plasma regime *Nat. Phys.* **18** 251–8
- [2] Gaffney J A, Brandon S T, Humbird K D, Kruse M K G, Nora R C, Luc Peterson J and Spears B K 2019 Making inertial confinement fusion models more predictive *Phys. Plasmas* **26** 082704
- [3] Döppner T *et al* 2018 Absolute equation-of-state measurement for polystyrene from 25 to 60 mbar using a spherically converging shock wave *Phys. Rev. Lett.* **121** 025001
- [4] Fernandez-Pañella A *et al* 2019 Shock compression of liquid deuterium up to 1 TPa *Phys. Rev. Lett.* **122** 255702
- [5] Kritcher A L *et al* 2020 A measurement of the equation of state of carbon envelopes of white dwarfs *Nature* **584** 51–54
- [6] Gaffney J A *et al* 2018 A review of equation-of-state models for inertial confinement fusion materials *High Energy Density Phys.* **28** 7–24
- [7] Mihaylov D I, Karasiev V V, Hu S X, Rygg J R, Goncharov V N and Collins G W 2021 Improved first-principles equation-of-state table of deuterium for high-energy-density applications *Phys. Rev. B* **104** 144104
- [8] Hu S X, Militzer B, Goncharov V N and Skupsky S 2011 First-principles equation-of-state table of deuterium for inertial confinement fusion applications *Phys. Rev. B* **84** 224109
- [9] Hu S X, Goncharov V N, Boehly T R, McCrory R L, Skupsky S, Collins L A, David Kress J and Militzer B 2015 Impact of first-principles properties of deuterium–tritium on inertial confinement fusion target designs *Phys. Plasmas* **22** 056304
- [10] Karasiev V V, Mihaylov D I and Hu S X 2022 Meta-GGA exchange-correlation free energy density functional to increase the accuracy of warm dense matter simulations *Phys. Rev. B* **105** L081109
- [11] Irwin Kerley G 2003 *Equations of state for hydrogen and deuterium* Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA (United States) SAND2003-3613 (<https://doi.org/10.2172/917468>)
- [12] Saumon D 2013 The sesame 5267 equation of state of deuterium *Technical Report* LA-UR-13-20032 (Los Alamos National Laboratory)
- [13] Danel J-F, Kazandjian L and Piron R 2016 equation of state of warm dense deuterium and its isotopes from density-functional theory molecular dynamics *Phys. Rev. E* **93** 043210
- [14] Becker A, Lorenzen W, Fortney J J, Nettelmann N, Schöttler M and Redmer R 2014 Ab initio equations of state for hydrogen (H-REOS.3) and helium (H-REOS.3) and their implications for the interior of brown dwarfs *Astrophys. J. Suppl. Ser.* **215** 21
- [15] Caillabet L, Mazevet S and Loubeyre P 2011 Multiphase equation of state of hydrogen from ab initio calculations in the range 0.2 to 5 g/cc up to 10 eV *Phys. Rev. B* **83** 094101
- [16] Fratanduono D E, Millot M, Fernandez Pañella A, Sterne P A, Collins G W, Hicks D G, Eggert J H, Boehly T R and Celliers P M 2019 Measurement of the sound speed in dense fluid deuterium along the cryogenic liquid hughoniot *Phys. Plasmas* **26** 012710
- [17] Knudson M D and Desjarlais M P 2017 High-precision shock wave measurements of deuterium: evaluation of exchange-correlation functionals at the molecular-to-atomic transition *Phys. Rev. Lett.* **118** 035501
- [18] He Z *et al* 2021 High-precision equation of state benchmark for cryogenic liquid deuterium at ultrahigh pressure *Phys. Rev. B* **103** 134107
- [19] Em Karniadakis G, Kevrekidis I G, Lu L, Perdikaris P, Wang S and Yang L 2021 Physics-informed machine learning *Nat. Rev. Phys.* **3** 422–40
- [20] Gaffney J A, Yang L and Ali S 2022 Constraining model uncertainty in plasma equation-of-state models with a physics-constrained gaussian process (arXiv:2207.00668)
- [21] Mentzer K L and Luc Peterson J 2023 Neural network surrogate models for equations of state *Phys. Plasmas* **30** 032704
- [22] Geman S, Bienenstock E and Doursat R 1992 Neural networks and the bias/variance Dilemma *Neural Comput.* **4** 1–58
- [23] Yang Z, Yu Y, You C, Steinhardt J and Ma Y 2020 Rethinking bias-variance trade-off for generalization of neural networks *Int. Conf. on Machine Learning* (PMLR) pp 10767–77
- [24] Garnelo M, Rosenbaum D, Maddison C, Ramalho T, Saxton D, Shanahan M, Whye Teh Y, Rezende D and Ali Eslami S M 2018 Conditional neural processes *Int. Conf. on Machine Learning* (PMLR) pp 1704–13
- [25] LeCun Y, Bengio Y and Hinton G 2015 Deep learning *Nature* **521** 436–44
- [26] Nguyen G, Dlugolinsky S, Bobák M, Tran V, Lopez Garcia Á, Heredia I, Malík P and Hluchý L 2019 Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey *Artif. Intell. Rev.* **52** 77–124
- [27] Sushma Reddy D and Rama Chandra Prasad P 2018 Prediction of vegetation dynamics using ndvi time series data and lstm *Model. Earth Syst. Environ.* **4** 409–19
- [28] Reichstein M, Camps-Valls G, Stevens B, Jung M, Denzler J and Carvalhais N 2019 Deep learning and process understanding for data-driven earth system science *Nature* **566** 195–204
- [29] Kashefi A, Rempe D and Guibas L J 2021 A point-cloud deep learning framework for prediction of fluid flow fields on irregular geometries *Phys. Fluids* **33** 027104
- [30] Raissi M, Perdikaris P and Karniadakis G E 2019 Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations *J. Comput. Phys.* **378** 686–707
- [31] Zhang D, Guo L and Em Karniadakis G 2020 Learning in modal space: solving time-dependent stochastic pdes using physics-informed neural networks *SIAM J. Sci. Comput.* **42** A639–65
- [32] Tartakovsky A M, Ortiz Marrero C, Perdikaris P, Tartakovsky G D and Barajas-Solano D 2020 Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems *Water Resour. Res.* **56** e2019WR026731
- [33] Yang L, Zhang D and Em Karniadakis G 2020 Physics-informed generative adversarial networks for stochastic differential equations *SIAM J. Sci. Comput.* **42** A292–317

- [34] Purja Pun G P, Batra R, Ramprasad R and Mishin Y 2019 Physically informed artificial neural networks for atomistic modeling of materials *Nat. Commun.* **10** 2339
- [35] Zhu Y, Zabaras N, Koutsourelakis P-S and Perdikaris P 2019 Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data *J. Comput. Phys.* **394** 56–81
- [36] Iten R, Metger T, Wilming H, Del Rio L and Renner R 2020 Discovering physical concepts with neural networks *Phys. Rev. Lett.* **124** 010508
- [37] Pfau D, Spencer J S, Matthews A G D G and Foulkes W M C 2020 Ab initio solution of the many-electron Schrödinger equation with deep neural networks *Phys. Rev. Res.* **2** 033429
- [38] Geneva N and Zabaras N 2022 Transformers for modeling physical systems *Neural Netw.* **146** 272–89
- [39] Bronstein M M, Bruna J, LeCun Y, Szlam A and Vandergheynst P 2017 Geometric deep learning: going beyond euclidean data *IEEE Signal Process. Mag.* **34** 18–42
- [40] Cohen T, Weiler M, Kiciranoğlu B and Welling M 2019 Gauge equivariant convolutional networks and the icosahedral cnn *Int. Conference on Machine Learning* (PMLR) pp 1321–30
- [41] Lu L, Jin P, Pang G, Zhang Z and Em Karniadakis G 2021 Learning nonlinear operators via deepnet based on the universal approximation theorem of operators *Nat. Mach. Intell.* **3** 218–29
- [42] Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart A and Anandkumar A 2020 Fourier neural operator for parametric partial differential equations (arXiv:2010.08895)
- [43] Pang G, Yang L and Em Karniadakis G 2019 Neural-net-induced gaussian process regression for function approximation and pde solution *J. Comput. Phys.* **384** 270–88
- [44] Reiser M and Burkhhardt H 2007 Learning equivariant functions with matrix valued kernels *J. Mach. Learn. Res.* **8** 385–408
- [45] Wang S, Wang H and Perdikaris P 2021 On the eigenvector bias of fourier feature networks: from regression to solving multi-scale pdes with physics-informed neural networks *Comput. Methods Appl. Mech. Eng.* **384** 113938
- [46] Raissi M, Perdikaris P and Em Karniadakis G 2018 Numerical gaussian processes for time-dependent and nonlinear partial differential equations *SIAM J. Sci. Comput.* **40** A172–98
- [47] Raissi M, Perdikaris P and Em Karniadakis G 2017 Machine learning of linear differential equations using gaussian processes *J. Comput. Phys.* **348** 683–93
- [48] Lagaris I E, Likas A and Fotiadis D I 1998 Artificial neural networks for solving ordinary and partial differential equations *IEEE Trans. Neural Netw.* **9** 987–1000
- [49] Bishop C M and Nasrabadi N M 2006 *Pattern Recognition and Machine Learning* vol 4 (Springer)
- [50] Aggarwal C C et al 2018 *Neural Networks and Deep Learning* vol 10 (Springer) p 3
- [51] Dietterich T 1995 Overfitting and undercomputing in machine learning *ACM Comput. Surv. (CSUR)* **27** 326–7
- [52] Zaheer M, Kottur S, Ravanbakhsh S, Poczos B, Salakhutdinov R and Smola A 2017 Deep sets (arXiv:1703.06114)
- [53] Kendall A and Gal Y 2017 What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems* p 30
- [54] Zhang S et al 2022 First-principles equation of state of chon resin for inertial confinement fusion applications *Phys. Rev. E* **106** 045207
- [55] Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L and Lerer A 2017 In NIPS Workshop Automatic differentiation in pytorch