



OPEN

# Uncertainty quantification in multivariable regression for material property prediction with Bayesian neural networks

Longze Li<sup>1</sup>, Jiang Chang<sup>1</sup>, Aleksandar Vakanski<sup>1✉</sup>, Yachun Wang<sup>2</sup>, Tiankai Yao<sup>3</sup> & Min Xian<sup>1</sup>

With the increased use of data-driven approaches and machine learning-based methods in material science, the importance of reliable uncertainty quantification (UQ) of the predicted variables for informed decision-making cannot be overstated. UQ in material property prediction poses unique challenges, including multi-scale and multi-physics nature of materials, intricate interactions between numerous factors, limited availability of large curated datasets, etc. In this work, we introduce a physics-informed Bayesian Neural Networks (BNNs) approach for UQ, which integrates knowledge from governing laws in materials to guide the models toward physically consistent predictions. To evaluate the approach, we present case studies for predicting the creep rupture life of steel alloys. Experimental validation with three datasets of creep tests demonstrates that this method produces point predictions and uncertainty estimations that are competitive or exceed the performance of conventional UQ methods such as Gaussian Process Regression. Additionally, we evaluate the suitability of employing UQ in an active learning scenario and report competitive performance. The most promising framework for creep life prediction is BNNs based on Markov Chain Monte Carlo approximation of the posterior distribution of network parameters, as it provided more reliable results in comparison to BNNs based on variational inference approximation or related NNs with probabilistic outputs.

**Keywords** Uncertainty quantification, Bayesian neural networks, Active learning, Creep life, Physics-informed machine learning

*Uncertainty Quantification* (UQ) plays a crucial role in various science and engineering disciplines. In the field of material science, the application of computational modeling methods has significantly accelerated the discovery of novel materials with enhanced properties. Determining the level of confidence in the predictions made by computational models is of high importance, as high levels of uncertainty can result in large deviations from the actual material behavior in practical applications<sup>1</sup>. With the increasing complexity of computational modeling, the computational cost associated with numerical UQ models has also increased, necessitating the development of computationally efficient methods for both prediction and uncertainty estimates<sup>2</sup>. In general, uncertainties can be categorized into two main types: aleatoric uncertainty, which arises due to the inherent process randomness (e.g., similarities in experimental data from the same experiment), and epistemic uncertainty, related to the discrepancies due to lack of training data or imperfections in the computational models<sup>3,4</sup>.

Recent advancements in Artificial Intelligence, particularly in Machine Learning (ML) and Neural Networks (NNs), ushered in a new era for design of experiments and materials modeling<sup>5</sup>. Among the *conventional ML models* with an inherent ability for UQ in regression tasks are Quantile Regression (QR)<sup>6</sup>, Gaussian Process Regression (GPR)<sup>7</sup>, and Natural Gradient Boosting (NGBoost)<sup>8</sup>. QR and NGBoost have shortcomings due to the lack of closed-form parameters estimation and are prone to overestimating the uncertainty level in data<sup>9</sup>. In most related works, GPR is generally reported as the state-of-the-art approach for UQ in multivariable regression and it stands out for its predictive accuracy and uncertainty estimates. On the other hand, GPR also has important limitations, since the commonly employed kernels, such as Gaussian and Matern kernels, have continuous sample paths, and are therefore less suitable for material properties prediction, where variations in microstructural

<sup>1</sup>Department of Computer Science, University of Idaho, Idaho Falls, ID 83404, USA. <sup>2</sup>Nuclear Science & Technology (NS&T), Idaho National Laboratory, Idaho Falls, ID 83415, USA. <sup>3</sup>Materials & Fuels Complex (MFC), Idaho National Laboratory, Idaho Falls, ID 83415, USA. ✉email: vakanski@uidaho.edu

features can have significant impact on macroscopic material properties<sup>10</sup>. Although researchers have proposed advanced GPR alternatives (e.g., sparse additive GPR<sup>11</sup>), they typically introduce novel challenges and require finetuning of additional hyperparameters.

In recent years, NNs have demonstrated remarkable success in both classification and regression tasks dealing with high-dimensional non-linear data<sup>12</sup>. Whereas NNs for classification tasks inherently output the probabilities in the model's prediction for each class, traditional NNs for regression tasks typically output only *single-point predictions* of the target variables (commonly referred to as *point estimates*). To provide uncertainty assessments, previous works proposed approaches such as Deep Ensembles<sup>13</sup> or Monte Carlo (MC) Dropout<sup>14</sup>. These methods introduced modifications in the outputs of traditional NNs with deterministic parameters to generate probabilistic predictions, allowing for UQ. Importantly, NNs with stochastic parameters, referred to as *Bayesian Neural Networks* (BNNs)<sup>15–17</sup>, have emerged as a promising approach for UQ that provides a probabilistic framework for capturing uncertainties in data-driven models. In comparison to conventional ML approaches for UQ, BNNs offer substantial flexibility in terms of the model structure, size, and parameter settings, and have the potential for efficient and reliable UQ modeling<sup>18</sup>.

In this work, we propose a novel approach for predicting creep rupture life in steel alloys using physics-informed BNNs. The approach incorporates physics-informed features based on governing creep laws into BNNs to estimate the uncertainties in the model's prediction of rupture life. The effectiveness of the proposed approach is additionally evaluated in the context of active learning (AL)<sup>19</sup>. By combining the variance reduction technique with *k*-mean clustering for selecting the most uncertain and diverse data points for training a model, we introduce a trade-off between exploration and exploitation of the solution space in AL. We conducted experimental validation with three datasets, consisting of collected data from creep tests with Stainless-Steel 316 alloys, Nickel-based superalloys, and Titanium alloys. The considered implementations of BNNs include networks employing Variational Inference (VI) and Markov Chain Monte Carlo (MCMC) approximation of the posterior distribution of the network parameters. We also evaluated the performance of traditional UQ regression models QR, NGBoost, and GPR, as well as deterministic NNs with point estimates and probabilistic outputs (Deep Ensembles, MC Dropout). The experimental results with a set of predictive single-point and uncertainty metrics demonstrate that MCMC BNNs are the most promising UQ method for creep rupture life prediction, with performance that is competitive with the performance of GPR. The results also demonstrate that physics-informed knowledge leverages the models' capacity for improved creep life prediction.

Although prior works have explored the application of ML approaches for predicting material properties<sup>20–26</sup> and uncertainty estimates in the predictions<sup>16,27–29</sup>, our proposed approach introduces novel concepts related to BNNs with incorporated physics priors for UQ in material property prediction. Specifically, the proposed work was inspired by Mamun et al.<sup>9</sup>, who proposed an approach for predicting the creep rupture life of ferritic steels using conventional ML methods. The authors employed GPR to calculate point estimates and uncertainty estimates in the predicted rupture life. Differently from the work by Mamun et al.<sup>9</sup>, we develop approaches for point regression and UQ in predicting creep rupture life based on BNNs, and demonstrate that Bayesian deep learning models present a promising framework for this task.

A body of work in the literature utilized Physics-Informed ML (PIML) to integrate knowledge from governing physics laws and data-driven methods for obtaining more consistent predictions<sup>30–34</sup>. In the proposed approach, we drew inspiration from the work by Zhang et al.<sup>31</sup>, where the authors introduced physics-informed features and a physics-informed NN loss for predicting creep rupture life. However, the authors did not consider UQ in their work, as well as, their focus was on designing standard NNs with deterministic parameters for creep life prediction. In another related paper, Olivier et al.<sup>16</sup> investigated the use of BNNs for UQ in the field of material science, and they developed VI-based ensemble methods for predicting the properties of composite materials. Differently from our work, the authors in<sup>16</sup> did not consider the integration of physics-informed knowledge in BNNs, and also they used simulated data as a proof-of-concept for the proposed approaches, whereas we used collected data from creep tests for experimental validation. Furthermore, researchers have proposed incorporating physics-informed priors into BNNs in prior works<sup>35,36</sup>, however, these works focus on other tasks, and to the best of our knowledge, this is the first work to apply such framework for material property prediction. Lastly, although many previous works have studied AL to prioritize the most informative sample for model training<sup>9,36–39</sup>, in this paper we show that physics-informed BNNs have the potential to accelerate the model training in AL for material property prediction.

The contributions of this paper include:

- Introduced a physics-informed BNNs approach for material property prediction, that incorporates physics knowledge for guiding the solutions of BNNs.
- Performed a comprehensive evaluation of the performance of conventional ML models, traditional NNs with probabilistic outputs, and BNNs for creep rupture life prediction and uncertainty estimation.
- Applied the UQ frameworks for an active learning task to iteratively select data points with the highest epistemic uncertainty and diversity for faster model training with fewer data points.

## Data and experimental setup

### Datasets

We use three datasets to validate the proposed UQ framework for creep rupture life prediction. The first creep dataset pertains to Stainless Steel (SS) 316 alloys, and it was obtained from the National Institute for Materials Science (NIMS) database<sup>40</sup>. The dataset contains 617 test samples with 20 features per sample. Specifically, the features provide information about the material composition related to the mass percent of the elements C, Si, Mn, P, S, Ni, Cr, Mo, Cu, Ti, Al, B, N, Nb, Ta and the material group of the alloy, testing conditions including

the applied stress (MPa) and temperature (°C), test measurements related to the percentage of elongation and percentage of area reduction, and the recorded creep rupture life (hours).

The second creep dataset is for Nickel-based superalloys and was adopted from the work by Han et al.<sup>41</sup>. The dataset includes 153 test samples with 15 features per sample. The features include material composition related to the weight percentage of the elements Ni, Al, Co, Cr, Mo, Re, Ru, Ta, W, Ti, Nb, and T, testing conditions including applied stress (MPa) and temperature (°C), and the recorded creep rupture life (hours).

The third creep dataset pertains to Titanium alloys and was adopted from Swetlana et al.<sup>42</sup>. It consists of 177 test samples with 24 features per sample. The features provide information about the weight percentage of the elements Ti, Al, V, Fe, C, H, O, Sn, Mb, Mo, Zr, Si, B, and Cr, testing conditions including applied stress (MPa) and temperature (°C), finishing conditions related to the solution treated temperature (°C), solution treated time (hours), annealing temperature (°C), annealing time (hours), test measurements of the steady-state strain rate (1/s) and strain to rupture (%), and the recorded creep rupture life (hours).

The reader can also refer to Supplementary Figs. S.1 to S.3 in the Supplementary Material file, which provide the histograms for all features in the three datasets, respectively, as well as additional statistics related to the average, minimum, maximum, and standard deviation values for the input features.

## Evaluation metrics

To evaluate the predictive accuracy of the models on unseen data samples, we used the following three metrics that are commonly used for regression models: coefficient of determination ( $R^2$ ), root-mean-squared error (RMSE), and mean absolute error (MAE). In addition, we used the Pearson Correlation Coefficient (PCC) to quantify the magnitude and direction of the linear relationship between the predicted values by the models and the target values.

To evaluate the quality of UQ we employed the following metrics: coverage and mean interval width.

*Coverage* quantifies the proportion of target values that fall within the predicted uncertainty interval by a regression model. Values of the coverage metric that are close to the nominal confidence interval of 95% are preferred for reliable uncertainty quantification. In some statistical works, the coverage metric is referred to as validity, since it assesses whether the predicted uncertainty intervals are valid.

*Mean interval width* calculates the average size of the predicted interval around the point estimates, related to the upper and lower uncertainty bounds. I.e., this metric assesses how tight the uncertainty bounds are across all predicted values. Smaller values of the interval width are preferred, as they indicate more precise uncertainty estimates by a regression model. In some statistical works, this metric is also referred to as sharpness.

## Methods

### Preliminaries: frameworks for uncertainty quantification

The considered problem is a multivariable regression task, where based on a set of  $N$  observed data samples  $\mathbf{X} = \{\mathbf{x}_i | \mathbf{x}_i \in \mathbb{R}^d, i = 1, 2, \dots, N\}$ , the goal is to estimate target values  $\mathbf{Y} = \{y_i | y_i \in \mathbb{R}, i = 1, 2, \dots, N\}$ . The observed data samples contain relevant material information, such as composition, known physical, mechanical, or other properties, and experimental conditions (such as temperature, and stress level) that are important for estimating the target variable.

For a training dataset  $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$  and a new data point  $\mathbf{x}^*$  which does not belong to the previously observed dataset, the objective is to find a mapping function  $f$  that estimates the target value, i.e.,  $y^* = f(\mathbf{x}^*)$ . The value  $y^*$  is referred to as *single-point prediction* or *point estimate*. In addition, the focus of this paper is on methods that provide *uncertainty quantification* for the predicted value  $y^*$ , either through a quantified measure of the variance of  $y^*$ , via confidence intervals, or by other means. The next sections provide an overview of conventional ML and DL frameworks for UQ in regression tasks.

### Conventional machine learning methods for UQ

Whereas many ML models for classification tasks inherently output the probabilities in the model's prediction for each class, most ML models for regression tasks typically provide only point estimates of the target value. Consequently, several approaches have been developed that employ or modify standard regression models in order to provide uncertainty estimates.

Quantile Regression (QR)<sup>6</sup> is a non-parametric approach for estimating the conditional quantiles—and therefore, uncertainties—in prediction variables. For a quantile  $\tau$ , the conditional quantile function of a target variable  $\mathbf{Y}$  given observed data  $\mathbf{X}$ ,  $Q_{\mathbf{Y}|\mathbf{X}}(\tau)$ , is calculated by minimizing a loss function. For a new data point  $\mathbf{x}^*$ , estimating the quantile functions  $Q_{y^*|\mathbf{x}^*}(\tau = 0.025)$  and  $Q_{y^*|\mathbf{x}^*}(\tau = 0.975)$  provides the 95% prediction interval of the target variable around the point estimate  $y^*$ . Advantages of QR include robustness to outliers, there are no distributional assumptions, and it can be used with any base model by replacing the original loss function with the quantile loss function.

Natural Gradient Boosting (NGBoost) Regression<sup>8</sup> is a probabilistic variant of the traditional Gradient Boosting method<sup>43</sup>. NGBoost method comprises three key components: a collection of base learners, parametric form of the conditional distribution  $\mathcal{P}(\mathbf{Y}|\mathbf{X}, \theta)$ , and scoring mechanism that ensures that the predicted distribution closely aligns with the actual distribution. For a new data point  $\mathbf{x}^*$ , the point estimate of the target variable  $y^*$  and the uncertainty quantified as the standard deviation  $\sigma^*$  are obtained from the conditional distribution  $\mathcal{P}(y^*|\mathbf{x}^*, \theta)$ . Advantages of NGBoost include the flexibility to be used with any base learners and any distributions with continuous parameters and scoring rules.

Gaussian Process Regression (GPR)<sup>7,44</sup> is a non-parametric Bayesian approach suitable for performing both function approximation and uncertainty estimation. Accordingly, GPR represents a collection of random

variables as a multivariate Gaussian distribution over a set of data points. For a Gaussian Process  $\mathcal{N}(\mathbf{Y}|\boldsymbol{\mu}, \mathbf{K})$ ,  $\mathbf{Y}$  denotes a vector of function values estimated at  $n$  data points  $\mathcal{Y} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)]$ ,  $\boldsymbol{\mu}$  is the mean of the Gaussian Process that by default is assigned to be zero, and  $\mathbf{K}$  is a positive definite covariance matrix. The smoothness of the distribution across functions is determined by the covariance kernel  $K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$  that defines the covariance between the function values  $f(\mathbf{x}_i)$  and  $f(\mathbf{x}_j)$ . Various kernel functions are used in practice, parameterized by a set of hyperparameters.

Given a training dataset  $(\mathbf{X}, \mathbf{Y})$ , for a new data point  $\mathbf{x}^*$  the posterior distribution of  $y^*$  is Gaussian, i.e.,  $\mathcal{P}(y^*|\mathbf{x}^*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\sigma}^{*2})$ . The mean of the predicted distribution  $\boldsymbol{\mu}^* = \mathbf{K}^{*T}(\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}$  is used as the point estimate of  $y^*$ , where  $\mathbf{K}^*$  is the covariance matrix between  $\mathbf{x}^*$  and the data points in the training dataset  $\mathbf{X}$ , and  $\sigma_n^2$  is the variance of independent and identically distributed (i.i.d.) Gaussian noise representing the uncertainty in the training data. The covariance of the predicted distribution is given with  $\boldsymbol{\sigma}^{*2} = k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}^{*T}(\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{K}^*$ . GPR is among the most powerful and flexible methods for UQ in regression tasks. By using different kernel functions and hyperparameters, GPR allows introducing domain knowledge and adapting the predictive distribution to the specific patterns and trends in a dataset.

#### Neural networks with deterministic parameters

Standard NNs with deterministic values of the parameters (weight and biases) have been increasingly employed for UQ of predicted values.

*Deep Ensembles* (DE)<sup>13</sup> involves training multiple NNs for a regression task, and aggregating their outputs to estimate the prediction uncertainties. The inherent randomness in the initializations of NN parameters and the associated training process drive the NNs to converge to different solutions in the hypothesis space. As a result, the DE approach results in samples of different network parameters that produce stochastic outputs<sup>45</sup>. For an ensemble of  $S$  NNs trained on the dataset  $(\mathbf{x}_i, y_i) \in \mathcal{D}$  and parameterized with parameters  $\theta_1, \theta_2, \dots, \theta_S$ , and for a new data point  $\mathbf{x}^*$ , the DE predictions are treated as a Gaussian distribution, where the predicted mean and standard deviation obtained by averaging the predictions of the ensemble are used as the target value and uncertainties estimates:

$$y^* = \frac{1}{S} \sum_{s=1}^S f(\mathbf{x}^*, \theta_s) \quad (1)$$

$$\sigma^* = \sqrt{\frac{1}{S} \sum_{s=1}^S (y^* - f(\mathbf{x}^*, \theta_s))^2} \quad (2)$$

*Monte Carlo (MC) Dropout*<sup>14</sup> is a simple extension of the standard dropout technique, which applies dropout during inference. For a new data point  $\mathbf{x}^*$ , MC Dropout performs  $M$  forward passes through the trained network with the dropout enabled to obtain Monte Carlo samples, resulting in  $M$  different predictions  $f(\mathbf{x}^*, \theta_i)$ . Similarly to Eqs. (1) and (2), the point and uncertainty estimates are computed from the resulting distribution of predicted target values  $f(\mathbf{x}^*, \theta_i)$ .

#### Neural networks with probabilistic parameters

Differently from standard NNs with deterministic parameters, Bayesian Neural Networks (BNNs) represent the network parameters with probability distributions, instead of fixed values. For a BNN model parameterized with parameters  $\theta$  that form probability distributions, inference for a new data point  $\mathbf{x}^*$  is performed by using the posterior predictive distribution  $\mathcal{P}(y^*|\mathbf{x}^*, \mathcal{D}) = \int \mathcal{P}(y^*|\mathbf{x}^*, \theta) \mathcal{P}(\theta|\mathcal{D}) d\theta$ . Direct calculation of the posterior distribution of the parameters given observed data  $\mathcal{P}(\theta|\mathcal{D})$  is intractable (and hence, the same applies to the predictive distribution  $\mathcal{P}(y^*|\mathbf{x}^*, \mathcal{D})$ ). Various approximations for  $\mathcal{P}(\theta|\mathcal{D})$  have been used in practice, among which the most popular methods are Variational Inference (VI) and Markov Chain Monte Carlo (MCMC).

*Variational Inference BNNs*: Variational Inference (VI) BNNs<sup>17</sup> employ an optimization technique to approximate the intractable posterior distribution  $\mathcal{P}(\theta|\mathcal{D})$  (that is,  $\mathcal{P}(\theta|\mathbf{X}, \mathbf{Y})$ ) with a simpler parameterized distribution  $q_\phi(\theta)$  (referred to as variational distribution) from a family of distributions  $\mathcal{Q}$ . The VI optimization is as follows:

$$\phi^* = \arg \min_{q_\phi(\theta) \in \mathcal{Q}} D_{\text{KL}}[q_\phi(\theta) || \mathcal{P}(\theta|\mathcal{D})], \quad (3)$$

where the goal is to calculate the parameters of the variational distribution  $q_\phi(\theta)$  that approximates the posterior distribution  $\mathcal{P}(\theta|\mathcal{D})$ . The Kullback–Leibler (KL) divergence is used as a measure of closeness between the two distributions. Directly calculating the KL divergence is also challenging, since it involves calculating the evidence  $\mathcal{P}(\mathbf{Y}|\mathbf{X})$ . To address this issue, an alternative approach has been developed, which utilizes the following Evidence Lower Bound (ELBO):

$$\text{ELBO}(\phi) = \mathbb{E}_{\theta \sim q_\phi(\theta)} [\log \mathcal{P}(\mathbf{Y}|\mathbf{X}, \theta)] - D_{\text{KL}}[q_\phi(\theta) || \mathcal{P}(\theta|\mathcal{D})]. \quad (4)$$

Since the KL divergence term  $D_{\text{KL}}[q_\phi(\theta) || \mathcal{P}(\theta|\mathcal{D})]$  in (3) is always non-negative, the expected log-likelihood of the data  $\log \mathcal{P}(\mathbf{Y}|\mathbf{X}, \theta)$  is always larger than ELBO. Therefore, using a loss function that maximizes ELBO in (4) minimizes the KL divergence between the variational distribution  $q_\phi(\theta)$  and the posterior distribution  $\mathcal{P}(\theta|\mathcal{D})$ .

The predictive uncertainty for new data point  $\mathbf{x}^*$  is estimated by sampling from the variational distribution  $q_\phi(\theta)$ . Using Eqs. (1) and (2), the point estimate and uncertainty are calculated as the mean and standard deviation of the drawn samples from  $q_\phi(\theta)$ .



*Markov Chain Monte Carlo (MCMC) BNNs:* MCMC BNNs<sup>46</sup> approximate the posterior distribution of NN parameters  $\theta$  given observational data  $\mathcal{P}(\theta|\mathbf{X}, \mathbf{Y})$  through Monte Carlo sampling. The approach employs a Markov Chain of model parameters, where each set of parameters  $\theta_i$  is a sample from the posterior distribution. To approximate the posterior distribution, the chain iteratively explores the space of possible parameters  $\theta_i$ . This exploration is guided by comparing the posterior probabilities, and as the Markov chain evolves, it effectively samples across the entire distribution space, allowing to converge to the target posterior distribution. After reaching a stationary distribution, for new input data point  $\mathbf{x}^*$ , the set of  $S$  generated samples  $\{\theta_1, \theta_2, \dots, \theta_S\}$  from  $\mathcal{P}(\theta|\mathbf{X}, \mathbf{Y})$  is used to generate  $S$  predictions  $f(\mathbf{x}^*, \theta_s)$ . Point estimates and uncertainty estimates are calculated by averaging the predictions, as in Eqs. (1) and (2).

Several MCMC sampling methods are used for approximating the posterior distribution with BNNs for regression tasks. Metropolis-Hasting algorithm<sup>47</sup> is often employed since it does not require exact knowledge about the probability distribution  $\mathcal{P}(\theta)$  to sample from, and a function that is proportional to the distribution is sufficient. Hamiltonian Monte Carlo (HMC) algorithm<sup>48</sup> is a version of Metropolis-Hasting that introduces a momentum term for proposing new states similar to simulating a physical system with Hamiltonian dynamics. Likewise, the No-U-Turn Sampling (NUTS) algorithm<sup>49</sup> is a sub-version of HMS that offers an approach for automatic selection of the hyperparameters.

### Physics-informed machine learning

Physics-Informed Machine Learning (PIML) integrates insights of the fundamental physics laws governing a process into ML models, in order to enhance the consistency of the predictions<sup>30–33,50,51</sup>. PIML can potentially address challenges associated with modeling material properties, as it leverages the demonstrated capability of ML methods—especially deep NNs—to capture intricate relationships within high-dimensional multi-scale and multi-physics data. Namely, accurately representing the dynamics and deformation mechanisms of materials with physics-based models (e.g., via partial differential equations) poses insurmountable difficulties, since it is exceptionally challenging to mathematically define all different underlying processes that change over time. Indeed, existing physics-based models capture only the most important factors that influence material properties, and are missing fine details of the underlying physics. PIML also can address the challenges posed by the limited availability of large curated datasets in materials science. Therefore, the fusion of historically collected material property data and physics-based models within a PIML framework holds the potential to enhance long-term predictions of material behavior under different conditions and exceeds the capabilities of traditional physics-based models.

In the proposed approach, we developed a PIML framework for predicting creep rupture life in metal alloys by introducing physics-informed feature engineering to augment the set of input features to the regression models and by applying a physics-informed loss function that introduces physics constraints into the learning algorithm.

#### *Physics-informed feature engineering*

We introduce two categories of physics-informed features based on estimations of the creep rupture life and stacking fault energy by using physics-based models.

*Creep Rupture Life Estimation:* Creep is a slow irreversible deformation process under the influence of stresses below the yield stress of a material. The prediction of creep rupture life, related to the time duration that a material can sustain before undergoing rupture is essential for guiding design strategies, maintenance regimens, and safety protocols for systems and structures. Existing physics-based creep models are broadly classified into two major categories: time–temperature parametric (TTP) models and creep constitutive (CC) models.

TTP models derive equations of thermal creep in metal materials by assuming interdependence between the effects of time and temperature on creep rupture life. The relationship between creep rupture life  $t_f$ , temperature  $T$ , and stress  $\sigma$  is established as  $\mathcal{P}(t_f, T) = f(\sigma)$ , where the function  $\mathcal{P}$  combines the rupture time and temperature into a single parameter. Well-known TTP models include the Larson–Miller method<sup>52</sup>  $\mathcal{P}(t_f, T) = T \cdot (C_{LM} + \log t_f)$ , Manson–Haferd method<sup>53</sup>  $\mathcal{P}(t_f, T) = (\log t_f - \log t_{in}) / (T - T_{in})$ , and Orr–Sherby–Dorn method<sup>54</sup>  $\mathcal{P}(t_f, T) = \log t_f - (Q_C / 2.3RT)$ . In these formulations,  $C_{LM}$  is a constant,  $Q_C$  is the creep activation energy,  $R$  denotes the universal gas constant, and  $t_{in}, T_{in}$  are constants that represent the point of intersection of the iso-stress lines in  $\log t_f$  versus  $T$  plots. The stress function  $f(\sigma)$  is typically represented as a cubic polynomial logarithmic function  $f(\sigma) = c_0 + c_1 \log \sigma + c_2 \log^2 \sigma + c_3 \log^3 \sigma$ , where the coefficients  $c_0, c_1, c_2, c_3$  are obtained via least-square regression fit to short-term experimental data. For a given material, based on establishing the relationship between the stress  $f(\sigma)$  versus the time–temperature parameter  $\mathcal{P}(t_f, T)$  from available short-term creep measurements, TTP methods extrapolate the plots to longer times to estimate the creep rupture life  $t_f$ .

Creep constitutive (CC) models are based on the Monkman–Grant (MG) conjecture<sup>55</sup>, which postulates that the product of the creep rupture life  $t_f$  and exponentiated minimum creep strain rate  $\dot{\epsilon}_{min}$  for isothermal conditions is constant. That is,  $t_f \cdot \dot{\epsilon}_{min}^n = C_{MG}$ , where  $n$  is the creep strain rate exponent, and  $C_{MG}$  is a constant. Researchers have proposed several CC models to enhance the creep life predictions. For instance, an alternative formulation of the creep strain rate is  $\dot{\epsilon} = a e^{-\frac{Q_{in}}{RT}} \cdot \sinh(b e^{-\frac{Q_{pw}}{RT}} \sigma)$ <sup>56</sup>, where the first term  $a e^{-\frac{Q_{in}}{RT}}$  describes the power-law creep mechanism in the high-stress range, and the term  $\sinh(b e^{-\frac{Q_{pw}}{RT}} \sigma)$  describes the viscous creep under the diffusion mechanism in moderate and low-stress ranges. The coefficients  $a$  and  $b$ , and the activation energies  $Q_{in}$  and  $Q_{pw}$  are obtained by fitting to experimental data of creep strain rate  $\dot{\epsilon}$  for given temperature  $T$  and stress  $\sigma$ . With the progress in material science, we can expect the development of novel models that more accurately predict creep behavior.

In this work, we employed the Manson–Haferd method for estimating the creep rupture life. The motivation is because the Manson–Haferd method was used for modeling the creep rupture life in SS316 alloys in the NIMS database, and the values of the coefficients  $c_0, c_1, c_2, c_3$  for the least-square regression fit are provided in the database. Accordingly, we used the Manson–Haferd method for estimating the creep rupture life for the Nickel-based superalloys and Titanium alloys datasets.

**Stacking fault energy (SFE):** SFE represents the energy difference between atoms within the regular lattice structure and those located in the stacking fault region. It defines how resistant a material is to deformation occurring along specific crystallographic planes. SFE is an important parameter that impacts the strength and deformation behavior of steel materials. To calculate SFE of stainless steels  $\gamma_{SFE}$ , we used the following equation<sup>57</sup>

$$\begin{aligned} \gamma_{SFE} = & \gamma^0 + 1.59Ni - 1.34Mn + 0.06Mn^2 - 1.75Cr + 0.01Cr^2 + 15.21Mo - 5.59Si \\ & - 60.69(C + 1.2N)^{\frac{1}{2}} + 26.27(C + 1.2N) * (C + 1.2Cr + Mn + Mo)^{\frac{1}{2}} \\ & + 0.6[Ni * (Cr + Mn)]^{1/2}, \end{aligned} \quad (5)$$

where SFE is approximated as a function of the chemical composition of the material where Ni, Mn, Cr, Mo, Si, C, and N denote weight percentages of the alloying elements, and  $\gamma^0$  is a constant equal to  $39mJ/m^2$  representing the SFE of pure austenitic iron at room temperature.

#### Physics-informed loss function

The PIML paradigm allows introducing initial, boundary conditions, and other types of physics constraints into the loss function of a learning algorithm. Motivated by the work of Zhang et al.<sup>31</sup>, we introduced two physics-informed boundary constraints regarding the predicted creep rupture life into the loss of NNs. The first introduced loss term is  $\mathcal{L}_{PI-B1} = \sum_{i=1}^N ReLU(y_i^*)$  which imposes that the predicted creep rupture life by the model  $y^*$  is non-negative. The second loss term is defined by  $\mathcal{L}_{PI-B2} = \frac{1}{N} \sum_{i=1}^N ReLU(a - y_i^*)$  that enforces that the predicted creep rupture life is upper bounded by a constant  $a$ . For the constant  $a$  we adopted the value of 100,000 h, because it is the greatest creep rupture life value in the three datasets. In these loss terms, ReLU denotes Rectified Linear Unit activation function, defined as  $ReLU(x) = \{0 \text{ for } x < 0, x \text{ for } x \geq 0\}$ . The two physics-informed terms regard negative and excessively large creep life values as physical violations that should be prevented from being output by the model. The two terms are added to the standard mean-squared error loss for regression tasks  $\mathcal{L}_{NN} = \frac{1}{N} \sum_{i=1}^N (y - y_i^*)^2$ , resulting in a composite physics-informed loss function of the model given by

$$\mathcal{L} = \mathcal{L}_{NN} + \lambda_1 \mathcal{L}_{PI-B1} + \lambda_2 \mathcal{L}_{PI-B2} \quad (6)$$

where  $\lambda_1$  and  $\lambda_2$  are weighting coefficients, which are empirically determined to quantify the contributions of the terms  $\mathcal{L}_{PI-B1}$  and  $\mathcal{L}_{PI-B2}$ , respectively. Specifically, we varied the values of the weight coefficients  $\lambda_1$  and  $\lambda_2$ , and we adopted values that resulted in the best overall performance on the three datasets.

#### Application case: active learning

Supervised ML requires that for each input data sample, there is an associated target value. Generally, for a supervised ML model to perform well, it often needs to be trained with a large number of labeled data points. In reality, labeled data may be scarce and expensive to obtain since the labeling or annotation process is time- and cost-consuming, whereas unlabeled data could often be accessed easily. Additionally, among the labeled data points, some could carry similar information, thus, contributing less value compared to the data points that carry dissimilar information. The Active Learning (AL) method was developed to select the most informative data points that speed up the training process<sup>59</sup>.

In pool-based AL, the training dataset  $\mathcal{D}$  comprises a pool of unlabeled data  $\mathcal{D}^U$  and labeled data  $\mathcal{D}^L$ . An initial ML model  $f$  is first trained with a small, randomly selected labeled dataset  $C \in \mathcal{D}^L$ . Next, a query strategy is applied to select the most informative data  $\mathcal{D}^A \in \mathcal{D}^U$  by employing an acquisition function to measure and rank the informativeness of the data. An annotator is asked to label the data  $\mathcal{D}^A$  selected by the query strategy, which is added to the dataset  $C$ , and the model is re-trained with the updated dataset. These steps are iteratively repeated until the model  $f$  converges<sup>37</sup>.

Two main types of query strategies are: uncertainty/informativeness-based strategies that ensure the informativeness of the unlabeled data, and representative-based/diversity-based strategies that measure the similarity of the instances and deal with issues such as sampling bias and inclusion of outliers. In addition, a hybrid query strategy combining these two strategies can also be applied<sup>37</sup>.

In this paper, we apply a hybrid query strategy that includes a Variance Reduction (VR) uncertainty-based method, and a  $k$ -means clustering diversity-based batch mode method to guide the selection of the newly added data<sup>9</sup>.

Variance reduction (VR) has been proven to be an effective AL method with regression tasks<sup>58</sup>, where the goal is to minimize the variance of the model. For a hypothesis  $f^{(C^{(i)})}$  learned on  $C$ , and a true hypothesis  $f^*$ , the total expectation of the error  $E_{out}(f^{(C^{(i)})}) = E_{\mathbf{x}}[(f^{(C^{(i)})}(\mathbf{x}) - f^*(\mathbf{x}))^2]$ , and the average estimation of learned hypotheses is  $\bar{f} \approx \frac{1}{K} \sum_{k=1}^K (f^{(C^{(k)})})$ , where  $k$  is a data point in  $C$ . The expectation on the generalization error of the entire dataset is  $E_D[E_{out}(f^{(C)})] = E_{\mathbf{x}}[Variance(\mathbf{x}) + bias(\mathbf{x})]$ . It indicates that for a given model, if the bias of the model is fixed, minimizing the variance of the model results in minimal generalization<sup>60</sup>. Therefore, VR selects and annotates the samples with the highest prediction variances, i.e., for which the model is most

uncertain of inferring with these samples. Adding these samples to the training dataset reduces the overall generalization error of the model<sup>36</sup>.

Conventional AL queries a single data point at each iteration, which is inefficient and leads to iterative training with small changes to the training dataset. To avoid these issues, we implemented Batch Mode Active Learning (BMAL)<sup>60</sup>. BMAL selects and annotates multiple samples  $B = \{B^{(1)}, B^{(2)}, \dots, B^{(m)}\} \in \mathcal{D}^U$  in each iteration and add these samples to the training dataset to re-train the model. On the other hand, an uncertainty-based BMAL query strategy may not be ideal since the most uncertain samples may be similar to each other. Therefore, diversity-based query strategy is typically preferred in BMAL. Clustering methods are commonly used to group the unlabeled dataset by similarity, and the most dissimilar samples are added to the training dataset. In this paper, we used  $k$ -means clustering to group the unlabeled samples into  $K$  clusters, where each group will have the least feature correlation with each other, and in each cluster the sample with the highest variance will be selected. Therefore,  $K$  number of samples will be selected and annotated for training.

## Experimental results

### Uncertainty quantification

For the SS316 alloys dataset, Table 1 presents the average and the standard deviation (in the subscript) of the metrics for predictive accuracy and uncertainty estimations for eight regression models for UQ based on five-fold cross-validation. High values of PCC indicate high correlations between the model predictions and the experimentally measured creep rupture life, and high values of  $R^2$  point to more consistent fit of the predicted values to the experimental creep rupture life. Similarly, low values of RMSE and MAE imply that the predicted creep rupture life more closely aligns with the experimentally measured creep rupture life. The results in Table 1 show that the standard NN model and BNNs performed better than the traditional ML models that include QR, NGBoost, and GPR. BNN-MCMC approach achieved the best performance for all point accuracy metrics, including PCC,  $R^2$ , RMSE, and MAE. In addition, BNN-MCMC produced the best results for the Interval width, except for the Coverage for which BNN-VI had the highest coverage. As expected, GPR achieved comparable performance to BNN-MCMC and it was the second best-performing method.

Figure 1 shows the target creep life, predicted creep life, and uncertainty estimates by the eight methods for one fold of the test dataset. The plots indicate that the uncertainty estimates by QR and NGBoost are over-estimated. On the other hand, GPR provides accurate uncertainty estimates, as well as the deterministic NNs and BNNs have generally good performance for both point predictions and uncertainty estimates.

The corresponding results for the Nickel-based superalloy dataset are shown in Table 2. Since this dataset is of smaller size and has only 153 data points, the performance decreased for all models. We can still note that BNN-MCMC, BNN-VI, and GPR have comparable predictions, and BNN-MCMC has a small advantage in point estimate predictions. The values of the uncertainty estimates indicate greater variability for all metrics due to the challenges associated with smaller datasets. The results for predicted creep rupture life and uncertainty estimates are shown in Fig. 2 only for the three best-performing models: GPR, BNN-VI, and BNN-MCMC.

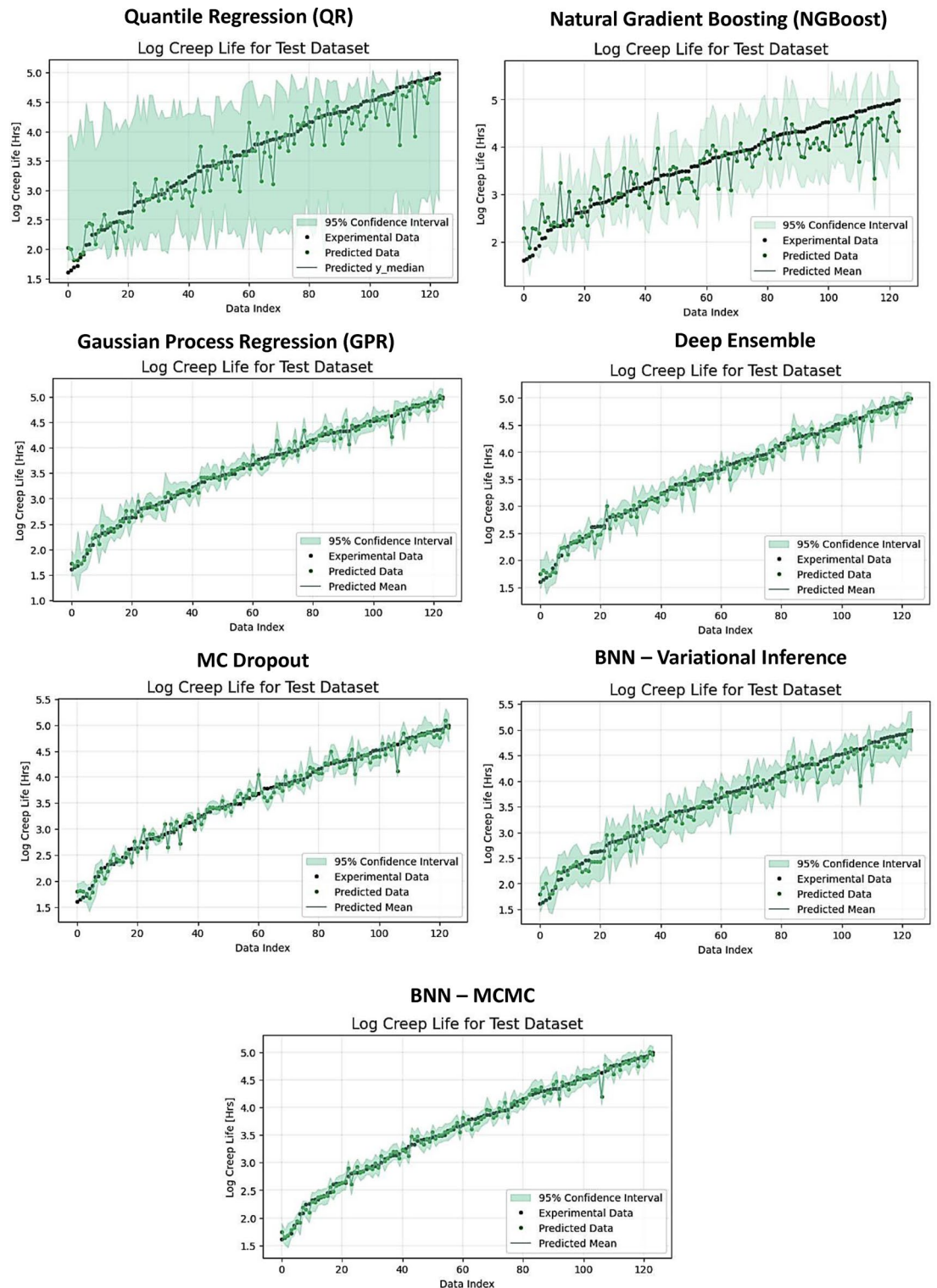
Next, we evaluated the models on the Titanium alloy dataset, which similar to the Nickel superalloys dataset is also much smaller than the SS316 dataset. The overall performance is consistent with the results in Tables 1 and 2, with the top performers being BNN-MCMC and GPR. The mean interval widths are wider for the top performers, indicating that the models are less confident about the uncertainty predictions. Figure 3 presents the predicted creep rupture life and uncertainty estimates for the three best-performing models: GPR, BNN-VI, and BNN-MCMC (Table 3).

### Physics-informed machine learning

Experimental validation encompasses standard NNs and three best-performing models from the previous section: GPR, BNN-VI, and BNN-MCMC. The models are compared to physics-informed (PI) variants. We introduced physics-informed features and a physics-informed loss function for the NN method. For the SS316 alloys dataset, the results are presented in Table 4. Noticeably, the PIML framework improves the points estimates

	PCC $\uparrow$	$R^2$ $\uparrow$	RMSE $\downarrow$ ( $\log_{10}h$ )	MAE $\downarrow$ ( $\log_{10}h$ )	Coverage $\uparrow$	Interval width $\downarrow$ ( $\log_{10}h$ )
NN	0.988 $_{\pm 0.004}$	0.973 $_{\pm 0.011}$	0.145 $_{\pm 0.031}$	1.027 $_{\pm 0.039}$		
QR	0.958 $_{\pm 0.011}$	0.913 $_{\pm 0.022}$	0.260 $_{\pm 0.028}$	0.195 $_{\pm 0.023}$	86.23 $_{\pm 3.13}$	1.98 $_{\pm 0.22}$
NGBoost	0.917 $_{\pm 0.025}$	0.826 $_{\pm 0.045}$	0.368 $_{\pm 0.042}$	0.13 $_{\pm 0.037}$	92.39 $_{\pm 1.87}$	1.34 $_{\pm 0.18}$
GPR	0.993 $_{\pm 0.001}$	0.987 $_{\pm 0.002}$	0.102 $_{\pm 0.007}$	0.072 $_{\pm 0.003}$	94.33 $_{\pm 2.54}$	0.39 $_{\pm 0.01}$
Deep Ensemble	0.991 $_{\pm 0.003}$	0.983 $_{\pm 0.006}$	0.115 $_{\pm 0.021}$	0.084 $_{\pm 0.018}$	87.20 $_{\pm 3.92}$	0.39 $_{\pm 0.08}$
MC Dropout	0.988 $_{\pm 0.002}$	0.977 $_{\pm 0.003}$	0.134 $_{\pm 0.012}$	0.101 $_{\pm 0.007}$	77.00 $_{\pm 3.69}$	0.33 $_{\pm 0.04}$
BNN-VI	0.984 $_{\pm 0.006}$	0.962 $_{\pm 0.014}$	0.172 $_{\pm 0.030}$	0.130 $_{\pm 0.017}$	<b>95.46</b> $_{\pm 2.49}$	0.69 $_{\pm 0.04}$
BNN-MCMC	<b>0.996</b> $_{\pm 0.001}$	<b>0.991</b> $_{\pm 0.001}$	<b>0.085</b> $_{\pm 0.008}$	<b>0.060</b> $_{\pm 0.004}$	92.71 $_{\pm 3.61}$	<b>0.30</b> $_{\pm 0.03}$

**Table 1.** Experimental results for the SS316 alloys dataset, including predictive accuracy metrics (PCC,  $R^2$ , RMSE, MAE) and uncertainty quantification metrics (coverage, interval width) for 8 compared methods: NN (Neural Network), QR (Quantile Regression), NGBoost (Natural Gradient Boosting), GPR (Gaussian Process Regression), Deep Ensemble, MC Dropout, BNN-VI (Variational Inference), and BNN-MCMC (Markov Chain Monte Carlo). Significant values are in bold.

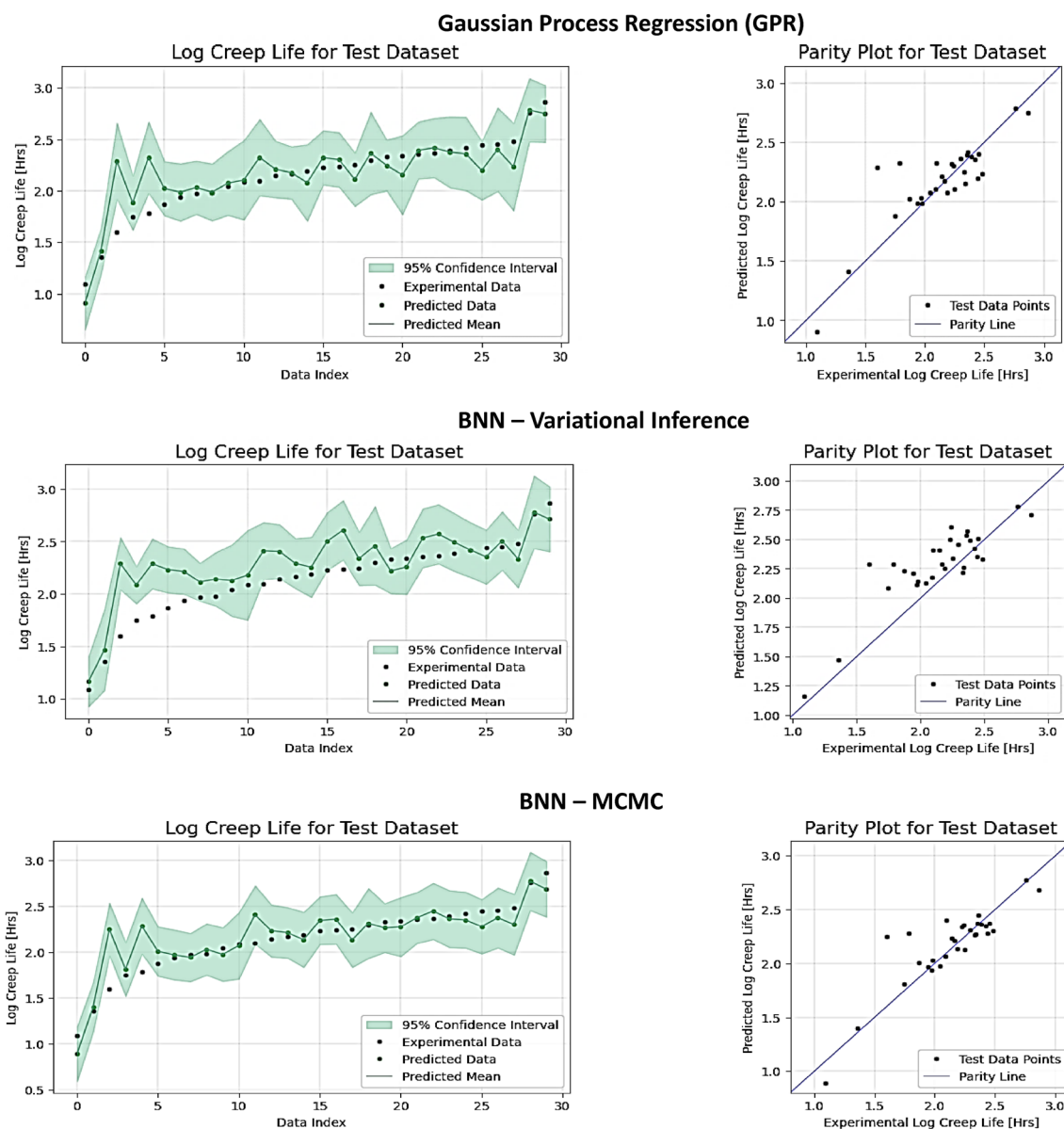


**Figure 1.** Experimental data points, predicted data points, and uncertainty estimates for Quantile regression (QR), Natural Gradient Boosting (NGBoost), Gaussian Process Regression (GPR), Deep Ensemble, MC Dropout, BNN-Variational Inference, and BNN-Markov Chain Monte Carlo (MCMC) for the SS316 alloys dataset. The logarithm values for the creep rupture life are shown on the vertical axis. The green shaded area represents the 95% confidence interval for the predictions by the models.



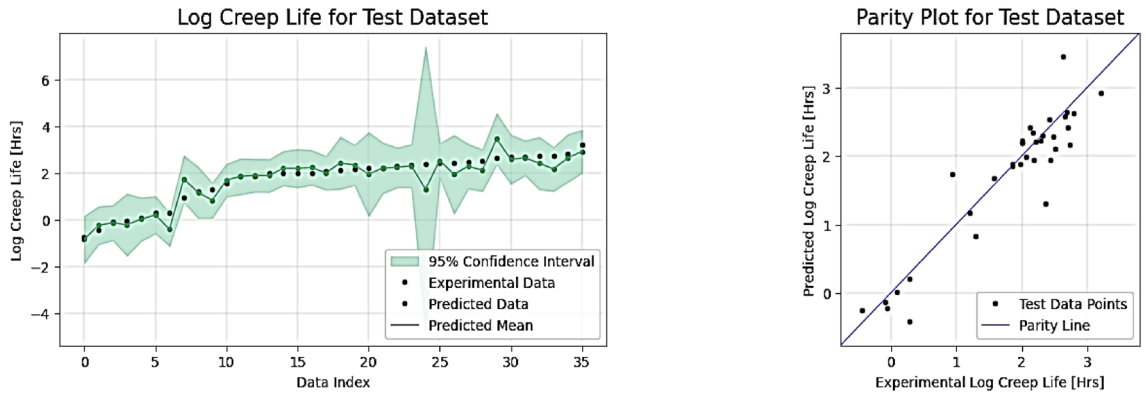
	PCC ↑	R <sup>2</sup> ↑	RMSE ↓ ( $\log_{10}h$ )	MAE ↓ ( $\log_{10}h$ )	Coverage ↑	Interval width ↓ ( $\log_{10}h$ )
NN	0.806 $\pm$ 0.113	0.620 $\pm$ 0.190	0.241 $\pm$ 0.030	0.427 $\pm$ 0.060		
QR	0.819 $\pm$ 0.048	0.641 $\pm$ 0.096	0.243 $\pm$ 0.032	0.191 $\pm$ 0.024	83.69 $\pm$ 8.66	0.997 $\pm$ 0.148
NGBoost	0.745 $\pm$ 0.101	0.554 $\pm$ 0.154	0.267 $\pm$ 0.035	0.213 $\pm$ 0.030	<b>95.41</b> $\pm$ 2.60	1.044 $\pm$ 0.170
GPR	0.907 $\pm$ 0.019	0.801 $\pm$ 0.054	0.175 $\pm$ 0.026	0.125 $\pm$ 0.011	92.77 $\pm$ 5.56	0.594 $\pm$ 0.019
Deep Ensemble	0.875 $\pm$ 0.074	0.748 $\pm$ 0.126	0.196 $\pm$ 0.032	0.143 $\pm$ 0.028	69.27 $\pm$ 10.6	0.414 $\pm$ 0.098
MC Dropout	0.858 $\pm$ 0.075	0.717 $\pm$ 0.149	0.207 $\pm$ 0.046	0.148 $\pm$ 0.027	45.72 $\pm$ 7.43	<b>0.209</b> $\pm$ 0.017
BNN-VI	0.909 $\pm$ 0.032	0.791 $\pm$ 0.069	0.185 $\pm$ 0.030	0.147 $\pm$ 0.025	87.52 $\pm$ 5.80	0.534 $\pm$ 0.029
BNN-MCMC	<b>0.914</b> $\pm$ 0.027	<b>0.824</b> $\pm$ 0.050	<b>0.167</b> $\pm$ 0.023	<b>0.116</b> $\pm$ 0.019	92.84 $\pm$ 4.71	0.607 $\pm$ 0.061

**Table 2.** Experimental results for the Nickel-based superalloys dataset, including predictive accuracy metrics (PCC, R<sup>2</sup>, RMSE, MAE) and uncertainty quantification metrics (coverage, interval width) for 8 compared methods: NN (Neural Network), QR (Quantile Regression), NGBoost (Natural Gradient Boosting), GPR (Gaussian Process Regression), Deep Ensemble, MC Dropout, BNN-VI (Variational Inference), and BNN-MCMC (Markov Chain Monte Carlo). Significant values are in bold.

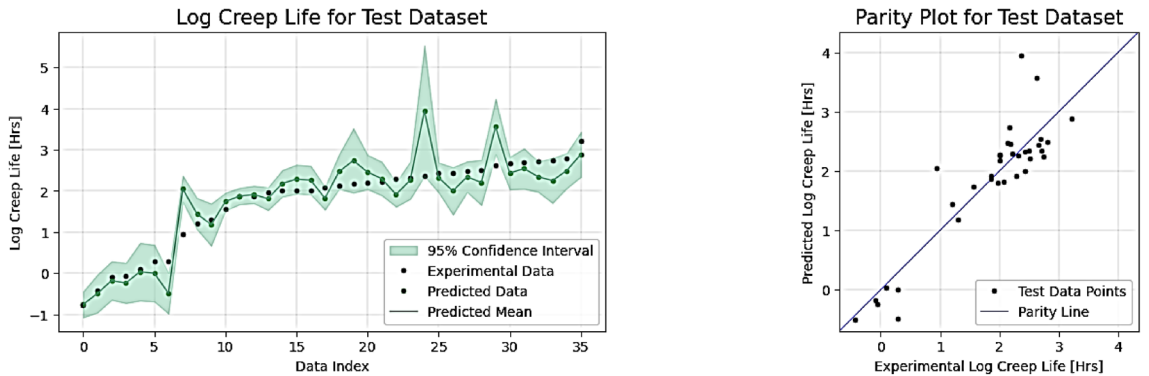


**Figure 2.** Experimental data points, predicted data points, and uncertainty estimates for Gaussian Process Regression (GPR), BNN-Variational Inference, and BNN-Markov Chain Monte Carlo (MCMC) for the Ni-based superalloys dataset. The logarithm values for the creep rupture life are shown on the vertical axis. The green shaded area represents the 95% confidence interval for the predictions by the models.

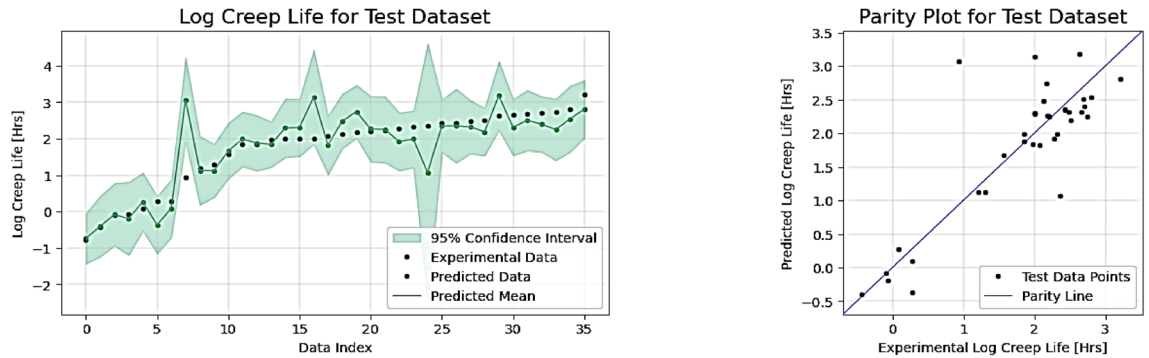
### Gaussian Process Regression (GPR)



### BNN – Variational Inference



### BNN – MCMC



**Figure 3.** Experimental data points, predicted data points, and uncertainty estimates for Gaussian Process Regression (GPR), BNN-Variational Inference, and BNN-Markov Chain Monte Carlo (MCMC) for the Ti-based alloys dataset. The logarithm values for the creep rupture life are shown on the vertical axis. The green shaded area represents the 95% confidence interval for the predictions by the models.

and uncertainty estimates for most regression models. Only for the PI-BNN-MCMC, the performance slightly decreased in comparison to the BNN-MCMC model without physics knowledge.

Table 5 presents the results for the Nickel-based superalloys dataset, and Table 6 shows the results for the Titanium alloys dataset. The incorporation of physics knowledge led to significant improvements in the predictions for all models for these two datasets. The best-performing approach is PI-BB-MCMC for both datasets. As expected, PIML imparts greater benefits to tasks with smaller datasets, and we can see higher gains for these two datasets, compared to the physics-informed models with the larger SS316 alloys dataset.

### Active learning

For the AL case study, we used a Batch Mode AL with a batch size  $B$  of 10 for the SS316 alloy dataset and a batch size  $B$  of 8 for the smaller Nickel-based superalloy and Titanium alloy datasets. Figure 4 plots the PCC values of the GPR, BNN-VI, and BNN-MCMC models. For the SS316 dataset, GPR achieves a higher PCC score with fewer data points. For Nickel and Titanium alloys, BNN MCMC achieves the best result, and it converges the fastest

	PCC $\uparrow$	R <sup>2</sup> $\uparrow$	RMSE $\downarrow$ ( $\log_{10}h$ )	MAE $\downarrow$ ( $\log_{10}h$ )	Coverage $\uparrow$	Interval width $\downarrow$ ( $\log_{10}h$ )
NN	0.794 $\pm$ 0.227	0.381 $\pm$ 0.777	0.800 $\pm$ 0.481	1.326 $\pm$ 0.150		
QR	0.584 $\pm$ 0.263	0.348 $\pm$ 0.320	0.396 $\pm$ 0.350	1.33 $\pm$ 0.099	81.98 $\pm$ 6.67	0.43 $\pm$ 0.19
NGBoost	0.510 $\pm$ 0.208	0.39 $\pm$ 0.395	0.453 $\pm$ 0.277	1.92 $\pm$ 0.073	94.90 $\pm$ 2.15	0.24 $\pm$ 0.75
GPR	0.921 $\pm$ 0.034	0.839 $\pm$ 0.066	0.471 $\pm$ 0.081	<b>0.309</b> $\pm$ 0.044	93.24 $\pm$ 2.22	1.99 $\pm$ 0.20
Deep Ensemble	0.903 $\pm$ 0.054	0.800 $\pm$ 0.109	0.517 $\pm$ 0.106	0.333 $\pm$ 0.041	79.10 $\pm$ 1.37	1.11 $\pm$ 0.171
MC Dropout	0.883 $\pm$ 0.101	0.641 $\pm$ 0.423	0.623 $\pm$ 0.345	0.348 $\pm$ 0.110	54.84 $\pm$ 5.62	0.51 $\pm$ 0.05
BNN-VI	0.919 $\pm$ 0.039	0.811 $\pm$ 0.116	0.474 $\pm$ 0.096	0.320 $\pm$ 0.040	75.10 $\pm$ 9.36	0.90 $\pm$ 0.06
BNN-MCMC	<b>0.922</b> $\pm$ 0.029	<b>0.843</b> $\pm$ 0.061	<b>0.449</b> $\pm$ 0.050	0.336 $\pm$ 0.023	<b>94.87</b> $\pm$ 4.21	1.77 $\pm$ 0.14

**Table 3.** Experimental results for the Titanium alloys dataset, including predictive accuracy metrics (PCC, R<sup>2</sup>, RMSE, MAE) and uncertainty quantification metrics (coverage, interval width) for 8 compared methods: NN (Neural Network), QR (Quantile Regression), NGBoost (Natural Gradient Boosting), GPR (Gaussian Process Regression), Deep Ensemble, MC Dropout, BNN-VI (Variational Inference), and BNN-MCMC (Markov Chain Monte Carlo). Significant values are in bold.

	PCC $\uparrow$	R <sup>2</sup> $\uparrow$	RMSE $\downarrow$ ( $\log_{10}h$ )	MAE $\downarrow$ ( $\log_{10}h$ )	Coverage $\uparrow$	Interval width $\downarrow$ ( $\log_{10}h$ )
NN	0.988 $\pm$ 0.004	0.973 $\pm$ 0.011	0.145 $\pm$ 0.031	<b>1.027</b> $\pm$ 0.039		
PI-NN	<b>0.992</b> $\pm$ 0.001	<b>0.982</b> $\pm$ 0.002	<b>0.120</b> $\pm$ 0.004	1.046 $\pm$ 0.037		
GPR	0.993 $\pm$ 0.001	0.987 $\pm$ 0.002	0.102 $\pm$ 0.007	0.072 $\pm$ 0.003	<b>94.33</b> $\pm$ 2.54	0.39 $\pm$ 0.01
PI-GPR	<b>0.995</b> $\pm$ 0.001	<b>0.990</b> $\pm$ 0.002	<b>0.091</b> $\pm$ 0.006	<b>0.064</b> $\pm$ 0.003	93.85 $\pm$ 2.07	<b>0.32</b> $\pm$ 0.01
BNN-VI	0.984 $\pm$ 0.006	0.962 $\pm$ 0.014	0.172 $\pm$ 0.030	0.130 $\pm$ 0.017	95.46 $\pm$ 2.49	0.69 $\pm$ 0.04
PI-BNN-VI	<b>0.993</b> $\pm$ 0.001	<b>0.984</b> $\pm$ 0.003	<b>0.113</b> $\pm$ 0.008	<b>0.087</b> $\pm$ 0.006	<b>95.95</b> $\pm$ 1.01	<b>0.49</b> $\pm$ 0.02
BNN-MCMC	<b>0.996</b> $\pm$ 0.001	<b>0.991</b> $\pm$ 0.001	<b>0.085</b> $\pm$ 0.008	<b>0.060</b> $\pm$ 0.004	92.71 $\pm$ 3.61	<b>0.30</b> $\pm$ 0.03
PI-BNN-MCMC	0.995 $\pm$ 0.001	0.990 $\pm$ 0.001	0.089 $\pm$ 0.002	0.064 $\pm$ 0.002	<b>93.19</b> $\pm$ 1.21	0.32 $\pm$ 0.01

**Table 4.** Experimental results for the SS316 alloys dataset with physics-informed features, including predictive accuracy metrics (PCC, R<sup>2</sup>, RMSE, MAE) and uncertainty quantification metrics (coverage, interval width) for 4 compared methods and their PI variants: NN (Neural Network), GPR (Gaussian Process Regression), BNN-VI (Variational Inference), and BNN-MCMC (Markov Chain Monte Carlo). Significant values are in bold.

	PCC $\uparrow$	R <sup>2</sup> $\uparrow$	RMSE $\downarrow$ ( $\log_{10}h$ )	MAE $\downarrow$ ( $\log_{10}h$ )	Coverage $\uparrow$	Interval width $\downarrow$ ( $\log_{10}h$ )
NN	0.806 $\pm$ 0.113	<b>0.620</b> $\pm$ 0.190	<b>0.241</b> $\pm$ 0.030	<b>0.427</b> $\pm$ 0.060		
PI-NN	<b>0.905</b> $\pm$ 0.042	0.130 $\pm$ 0.470	0.356 $\pm$ 0.139	0.556 $\pm$ 0.081		
GPR	<b>0.907</b> $\pm$ 0.019	0.801 $\pm$ 0.054	0.175 $\pm$ 0.026	0.125 $\pm$ 0.011	92.77 $\pm$ 5.56	0.594 $\pm$ 0.019
PI-GPR	0.905 $\pm$ 0.011	<b>0.933</b> $\pm$ 0.021	<b>0.102</b> $\pm$ 0.006	<b>0.074</b> $\pm$ 0.007	<b>94.73</b> $\pm$ 1.63	<b>0.46</b> $\pm$ 0.05
BNN-VI	0.909 $\pm$ 0.032	0.791 $\pm$ 0.069	0.185 $\pm$ 0.030	0.147 $\pm$ 0.025	87.52 $\pm$ 5.80	0.534 $\pm$ 0.029
PI-BNN-VI	<b>0.972</b> $\pm$ 0.009	<b>0.936</b> $\pm$ 0.015	<b>0.101</b> $\pm$ 0.011	<b>0.071</b> $\pm$ 0.007	<b>97.38</b> $\pm$ 2.43	<b>0.52</b> $\pm$ 0.02
BNN-MCMC	0.914 $\pm$ 0.027	0.824 $\pm$ 0.050	0.167 $\pm$ 0.023	0.116 $\pm$ 0.019	92.84 $\pm$ 4.71	0.607 $\pm$ 0.061
PI-BNN-MCMC	<b>0.992</b> $\pm$ 0.004	<b>0.983</b> $\pm$ 0.009	<b>0.049</b> $\pm$ 0.007	<b>0.036</b> $\pm$ 0.003	<b>93.46</b> $\pm$ 5.80	<b>0.20</b> $\pm$ 0.03

**Table 5.** Experimental results for the Nickel-based superalloys dataset with physics-informed features, including predictive accuracy metrics (PCC, R<sup>2</sup>, RMSE, MAE) and uncertainty quantification metrics (coverage, interval width) for 4 compared methods and their PI variants: NN (Neural Network), GPR (Gaussian Process Regression), BNN-VI (Variational Inference), and BNN-MCMC (Markov Chain Monte Carlo). Significant values are in bold.

with a small number of data samples. BNN-MCMC performed the best in 2 of the 3 tests, and GPR performed the best in 1 of the tests, whereas BNN-VI had the lowest performance overall.

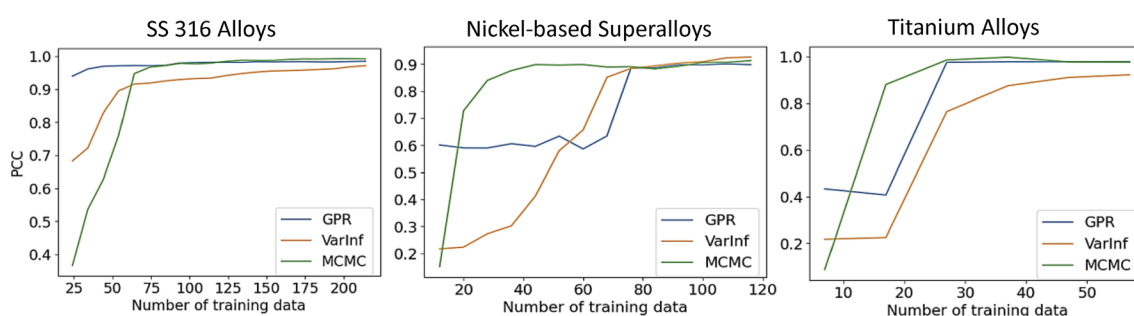
### Implementation details

In the experiments, creep rupture life is used as the target variable in the regression models, and the remaining features in the datasets are used as inputs to the models. The input features are normalized to the range between 0 and 1, and base 10 logarithm transformation is applied to the values of the creep rupture life. Five-fold cross-validation is used to report the performance of different models.

For the implementation of QR, NGBoost, and GPR, we adopted the same hyperparameters as in the work by Mamun et al.<sup>9</sup> Following their work, we used CatBoost model with QR for calculating the quantiles, and we

	PCC $\uparrow$	R2 $\uparrow$	RMSE $\downarrow$ ( $\log_{10}h$ )	MAE $\downarrow$ ( $\log_{10}h$ )	Coverage $\uparrow$	Interval width $\downarrow$ ( $\log_{10}h$ )
NN	0.794 $\pm$ 0.227	<b>0.381</b> $\pm$ 0.777	<b>0.800</b> $\pm$ 0.481	<b>1.326</b> $\pm$ 0.150		
PI-NN	<b>0.851</b> $\pm$ 0.095	- 0.119 $\pm$	1.079 $\pm$ 1.723	1.606		
GPR	0.921 $\pm$ 0.034	0.839 $\pm$ 0.066	0.471 $\pm$ 0.081	0.309 $\pm$ 0.044	93.24 $\pm$ 2.22	1.99 $\pm$ 0.20
PI-GPR	<b>0.941</b> $\pm$ 0.027	<b>0.877</b> $\pm$ 0.054	<b>0.433</b> $\pm$ 0.118	<b>0.298</b> $\pm$ 0.076	<b>93.46</b> $\pm$ 6.85	<b>1.82</b> $\pm$ 0.31
BNN-VI	0.919 $\pm$ 0.039	0.811 $\pm$ 0.116	<b>0.474</b> $\pm$ 0.096	<b>0.320</b> $\pm$ 0.040	<b>75.10</b> $\pm$ 9.36	<b>0.90</b> $\pm$ 0.06
PI-BNN-VI	<b>0.934</b> $\pm$ 0.015	<b>0.836</b> $\pm$ 0.051	0.503 $\pm$ 0.063	0.353 $\pm$ 0.027	73.16 $\pm$ 5.37	0.96 $\pm$ 0.05
BNN-MCMC	0.922 $\pm$ 0.029	0.843 $\pm$ 0.061	<b>0.449</b> $\pm$ 0.050	0.336 $\pm$ 0.023	<b>94.87</b> $\pm$ 4.21	1.77 $\pm$ 0.14
PI-BNN-MCMC	<b>0.937</b> $\pm$ 0.012	<b>0.865</b> $\pm$ 0.031	0.465 $\pm$ 0.082	<b>0.288</b> $\pm$ 0.029	86.90 $\pm$ 6.32	<b>1.09</b> $\pm$ 0.32

**Table 6.** Experimental results for the Titanium alloys dataset with physics-informed features, including predictive accuracy metrics (PCC, R2, RMSE, MAE) and uncertainty quantification metrics (coverage, interval width) for 4 compared methods and their PI variants: NN (Neural Network), GPR (Gaussian Process Regression), BNN-VI (Variational Inference), and BNN-MCMC (Markov Chain Monte Carlo). Significant values are in bold.



**Figure 4.** PCC values for BMAL case with Gaussian Process Regression (GPR), BNN-Variational Inference, and BNN-Markov Chain Monte Carlo (MCMC) for SS316 alloys, Nickel-based superalloys, and Titanium alloys.

applied the same kernel functions for GPR as in Mamun et al.<sup>9</sup> We used the scikit-learn<sup>61</sup>, ngboost, and catboost libraries for model training and evaluation.

For the NN networks, we performed extensive fine-tuning of the hyperparameters of all models. We conducted a random search over the hyperparameters of the different networks, which included search over the number of layers, number of neurons per layer, learning rate, batch size, number of epochs, dropout rate in dropout layers, and type of optimization algorithm. We used multiple-fold cross-validation for hyperparameter finetuning and model selection.

For implementing the BNN-VI method, the architecture consists of two fully-connected layers with 100 neurons in each layer. We applied Rectified Linear Unit (ReLU) activation function to the output of each layer. For the prior distribution of the network parameters, we adopted a normal distribution with a mean of 0 and a standard deviation of 0.06. The loss function is based on Eq. (6), with the weight coefficient for the KL divergence term set to 0.01. We used a Stochastic Gradient Descent (SGD) optimizer with Nesterov Momentum set to 0.95, and the learning rate was set to 0.001. After training the model, for inference we generated 1000 samples from the variational distribution. The point estimates and uncertainty estimates are calculated as the mean and  $\pm 3$  standard deviations of the drawn samples, according to Eqs. (1) and (2).

For the BNN-MCMC approach, we selected an architecture with three fully-connected layers with 10 neurons in each layer. For the prior distribution of the network parameters, we adopted a normal distribution with a mean of 0 and a standard deviation of 1. For approximating the posterior distribution we used the No-U-Turn Sampling (NUTS)<sup>49</sup> algorithm for training the model. For inference, we drew 100 samples, and the point estimates and uncertainty estimates were calculated similarly to the BNN-VI methods by taking the arithmetic mean and  $\pm 3$  standard deviations of the generated samples.

For the Deep Ensemble method, we used an ensemble of 5 base learners, each of which is a standard NN with three fully-connected layers having 10 neurons in each layer. Each hidden layer is followed with a dropout layer with a rate of 0.5 and a ReLU activation layer. Mean-square error (MSE) loss function was used, and the Adaptive Moment Estimation (Adam) optimizer with a learning rate of 0.01 was selected for training the models. The final predictions were calculated by taking the arithmetic mean and  $\pm 3$  standard deviations from the outputs of the base learners.

For the MC Dropout approach, we used an NN with three fully-connected layers having 100 neurons in each layer. Similar to the Deep Ensemble, we used a dropout rate of 0.5, ReLU activation function, MSE loss, and Adaptive Gradient Algorithm (Adagrad) optimizer with a learning rate of 0.01. For inference, we generated 1000 predictions, which were used to calculate the point estimates and uncertainty estimates.



For comparison, we implement standard NNs with deterministic parameters, consisting of three fully-connected layers with 1000, 200, and 40 neurons, respectively. We used ReLU activation function after each hidden layer. The loss function was MSE, and the Root Mean Squared Propagation (RMSprop) optimizer was used with a learning rate of 0.01.

BNN-MCMC, BINN-VI, Deep Ensemble, and MC Dropout were implemented using the PyTorch library<sup>62</sup>. For building BNN-MCMC we used the Pyro library<sup>63</sup>, and BNN-VI was implemented with the torchbnn library. For the above models, we chose a batch size of 16 data points.

## Discussion

The experimental results in Tables 1, 2 and 3 indicate that BNN models and GPR outperform the traditional ML models, and therefore they are more suitable for UQ in material property prediction. Overall, BNN-MCMC achieved comparable performance to GPR on most metrics, with a marginal increase in PCC of 0.004 and  $R^2$  of 0.010 and a marginal decrease in RSME of 0.016 and in MAE of 0.002. The results in Tables 4, 5 and 6 demonstrate that the PIML paradigm can improve the models' performance for the smaller Nickel-based superalloy and Titanium alloy datasets, whereas it obtained smaller improvement for the larger SS316 alloys dataset. For the application of UQ in AL, BNN-MCMC and GPR performed comparatively well.

The experimental results demonstrate the potential of BNNs to provide accurate and reliable UQ. Bayesian NNs exhibit more consistent uncertainty estimates that align better with the observed deviations, reducing the likelihood of overconfidence or underconfidence. Additionally, the priors in BNNs can be regarded as soft constraints that act similar to the regularization techniques in traditional NNs. Likewise, physics-informed BNNs leverage physical knowledge to constrain and guide the learning process, and AL actively selects the most informative samples to ensure the models learn faster with fewer samples.

One major limitation of BNNs is their high computational cost, since both BNN-MCMC and BNN-VI require sampling from the posterior distribution over the network parameters. Furthermore, MCMC requires sufficient number of iterations to obtain accurate samples from the posterior distribution, which can take significantly longer than GPR and require increased computational resources. Also, hyperparameter tuning of BNNs can be challenging and requires a deeper understanding of the internal working of the models.

It is also worth noting that GPR primarily focuses on modeling structural uncertainty reflecting the inherent variability within the model structure, as the function space is modeled as a distribution over functions, and the predictions are made by considering the possible functions that are consistent with the observed data. On the other hand, BNNs are aimed at quantifying parametric uncertainty that arises from the variability in the model parameters, rather than uncertainty in the functional form of the model.

A limitation of this work is that the collected data from creep rupture tests do not comprise data from repeated experimental measurements that capture the variability in the measured creep life values. Consequently, the used datasets do not provide ground truth values to allow for truthful evaluation and comparison of the used methods for uncertainty quantification.

In future work, we will focus on the development of physics-informed loss functions and physics-informed layers in BNNs, and on the implementation of AL approaches based on hybrid query strategies, such as Query by Committee.

## Conclusion

This work provides a study of uncertainty quantification in multivariable regression for material property prediction with Bayesian Neural Networks. We employed the Bayesian framework for predicting creep rupture life in metal alloys. Our findings demonstrate the potential of BNNs to advance the field of materials science and engineering by enabling more accurate and reliable predictions with quantified uncertainties. The experimental validation indicates that the most promising NN approach for material property prediction is BNN-MCMC, which achieved performance that is competitive with the performance of GPR as the state-of-the-art method for UQ is multivariable regression. The case study on applying uncertainty estimates in an active learning scenario confirms that BNN is a promising approach for overcoming the challenges in modeling material properties related to sparse and noisy data.

## Data availability

The data used in this study are available upon request to Aleksandar Vakanski (vakanski@uidaho.edu).

Received: 13 December 2023; Accepted: 2 May 2024

Published online: 08 May 2024

## References

1. Wang, Y., McDowell, D. L., eds. *Uncertainty Quantification in Multiscale Materials Modeling* (Woodhead Publishing, 2020).
2. Smith, R. C. *Uncertainty Quantification: Theory, Implementation, and Applications* (Siam Publications Library, 2013).
3. Abdar, M. *et al.* A review of uncertainty quantification in deep learning: Techniques, applications, and challenges. *Inf. Fusion* **76**, 243–297 (2021).
4. Acar, P. Recent progress of uncertainty quantification in small-scale materials science. *Prog. Mater. Sci.* **117**, 100723 (2021).
5. Choudhary, K. *et al.* Recent advances and applications of deep learning methods in materials science. *Comput. Mater.* **8**, 59 (2022).
6. Koenker, R. & Hallock, K. F. Quantile regression. *J. Econ. Perspect.* **15**, 143–156 (2001).
7. Williams, C. K., Rasmussen, C. E. Gaussian processes for regression. In *Advances in Neural Information Processing Systems* (1995).
8. Duan, T., Avati, A., Ding, D. Y., Thai, K. K., Basu, S., Ng, A. Y., Schuler, A. NGBoost: Natural gradient boosting for probabilistic prediction. In *International Conference on Machine Learning* (2020).
9. Mamun, O., Taufique, M. F. N., Wenzlick, M., Hawk, J. & Devanathan, R. Uncertainty quantification for Bayesian active learning in rupture life prediction of ferritic steels. *Sci. Rep.* **12**, 2083 (2022).

10. Lei, B. *et al.* Bayesian optimization with adaptive surrogate models for automated experimental design. *Comput. Mater.* **7**, 194 (2021).
11. Luo, H., Nattino, G. & Pratola, M. T. Sparse additive Gaussian process regression. *J. Mach. Learn. Res.* **23**, 2652–2685 (2022).
12. Bengio, Y., LeCun, Y. & Hinton, G. Deep learning for AI. *Commun. ACM* **64**, 58–65 (2021).
13. Lakshminarayanan, B., Pritzel, A., Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems* (2017).
14. Gal, Y., Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning* (2016).
15. Neal, R. M. *Bayesian Learning for Neural Networks* (Springer Science & Business Media, 2012).
16. Olivier, A., Shields, M. D. & Graham-Brady, L. Bayesian neural networks for uncertainty quantification in data-driven materials modeling. *Comput. Methods Appl. Mech. Eng.* **386**, 114079 (2021).
17. Graves, A. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems* (2011).
18. Snoek, J. *et al.* Scalable Bayesian optimization using deep neural networks. In *International Conference on Machine Learning* (2015).
19. Settles, B. Active learning literature survey. (2009).
20. Biswas, S., Castellanos, D. F. & Zaiser, M. Prediction of creep failure time using machine learning. *Sci. Rep.* **10**, 16910 (2020).
21. Mamun, O., Wenzlick, M., Sathanur, A., Hawk, J. & Devanathan, R. Machine learning augmented predictive and generative model for rupture life in ferritic and austenitic steels. *Mater. Degrad.* **5**, 20 (2021).
22. Mamun, O., Wenzlick, M., Hawk, J. & Devanathan, R. A machine learning aided interpretable model for rupture strength prediction in Fe-based martensitic and austenitic alloys. *Sci. Rep.* **11**, 5466 (2021).
23. Hossein, M. A. & Stewart, C. M. A probabilistic creep model incorporating test condition, initial damage, and material property uncertainty. *Int. J. Press. Vessels Piping* **193**, 104446 (2021).
24. Liang, M. *et al.* Interpretable ensemble-machine-learning models for predicting creep behavior of concrete. *Cem. Concr. Compos.* **125**, 104295 (2022).
25. Zhang, X. C., Gong, J. G. & Xuan, F. Z. A deep learning based life prediction method for components under creep, fatigue and creep-fatigue conditions. *Int. J. Fatigue* **148**, 106236 (2021).
26. Gu, H. H. *et al.* Machine learning assisted probabilistic creep-fatigue damage assessment. *Int. J. Fatigue* **156**, 106677 (2022).
27. Gawlikowski, J. *et al.* A survey of uncertainty in deep neural networks. *Artif. Intell. Rev.* **56**, 1513–1589 (2021).
28. Hirschfeld, L. *et al.* Uncertainty quantification using neural networks for molecular property prediction. *J. Chem. Inf. Model.* **60**, 3770–3780 (2020).
29. Zhang, P., Yin, Z. Y. & Jin, Y. F. Bayesian neural network-based uncertainty modelling: Application to soil compressibility and undrained shear strength prediction. *Can. Geotech. J.* **59**, 546–557 (2022).
30. Karniadakis, G. *et al.* Physics-informed machine learning. *Nat. Rev. Phys.* **3**, 422–440 (2021).
31. Zhang, X. C., Gong, J. G. & Xuan, F. Z. A physics-informed neural network for creep-fatigue life prediction of components at elevated temperatures. *Eng. Fract. Mech.* **258**, 108130 (2021).
32. Salvati, E. *et al.* A defect-based physics-informed machine learning framework for fatigue finite life prediction in additive manufacturing. *Mater. Des.* **222**, 111089 (2022).
33. Chen, D., Li, Y., Liu, K. & Li, Y. A physics-informed neural network approach to fatigue life prediction using small quantity of samples. *Int. J. Fatigue* **166**, 107270 (2023).
34. Paszke, A., Gross, S., Massa, F. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **32** (2019).
35. Yang, L., Meng, X. & Karniadakis, G. B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *J. Comput. Phys.* **425**, 109913 (2021).
36. Perez, S., Maddu, S., Sbalzarini, I. F. & Poncet, P. Adaptive weighting of Bayesian physics informed neural networks for multitask and multiscale forward and inverse problems. *J. Comput. Phys.* **491**, 112342 (2023).
37. Cohn, D., Ghahramani, Z. & Jordan, M. Active learning with statistical models. *J. Artif. Intell. Res.* **4**, 129–145 (1996).
38. Fu, Y., Zhu, Z. & Li, B. A survey on instance selection for active learning. *Knowl. Inf. Syst.* **35**, 249–283 (2013).
39. Schein, A. I. & Ungar, L. H. Active learning for logistic regression: An evaluation. *Mach. Learn.* **68**, 235–265 (2007).
40. NIMS Materials Database (MatNavi)—DICE, National Institute for Materials Science. *Mits.nims.go.jp* (accessed 16 Oct 2023).
41. Han, H., Li, W., Antonov, S. & Li, L. Mapping the creep life of nickel-based SX superalloys in a large compositional space by a two-model linkage machine learning method. *Comput. Mater. Sci.* **205**, 111229 (2022).
42. Svetlana, S., Rout, A. & Kumar Singh, A. Machine learning assisted interpretation of creep and fatigue life in titanium alloys. *APL Mach. Learn.* **1**, 016102 (2023).
43. Friedman, J. H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **29**, 1189–1232 (2001).
44. Schulz, E., Maarten, S. & Andreas, K. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *J. Math. Psychol.* **85**, 1–16 (2018).
45. Ganaie, M. A. *et al.* Ensemble deep learning: A review. *Eng. Appl. Artif. Intell.* **115**, 105151 (2022).
46. Gamerman, D., Lopes, H. F. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference* (CRC Press, 2006).
47. Chib, S. & Greenberg, E. Understanding the Metropolis-Hastings algorithm. *Am. Stat.* **49**, 327–335 (1995).
48. Girolami, M. & Calderhead, B. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **73**, 123–214 (2011).
49. Hoffman, M. D. & Gelman, A. The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.* **15**, 1593–1623 (2014).
50. Lagaris, I. E., Likas, A. & Fotiadis, D. I. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans. Neural Netw.* **9**(5), 987–1000 (1998).
51. Yu, R., Perdikaris, P., Karpapne. A. Physics-guided ai for large-scale spatiotemporal data. *Knowl. Discov. Data Min.* 4088–4089 (2021).
52. Larson, F. R. & Miller, J. A time-temperature relationship for rupture and creep stresses. *J. Fluids Eng.* **74**, 765–775 (1952).
53. Manson, S. S., Haferd, A. M. A linear time-temperature relation for extrapolation of creep and stress-rupture data, NASA-TN-2890 (1953).
54. Orr, R., Sherby, O. & Dorn, J. Correlations of rupture data for metals at elevated temperatures. *Trans. Am. Soc. Metals* **46**, 113–156 (1953).
55. Monkman, F. C. & Grant, N. J. The Monkman–Grant relationship. *Proc. Am. Soc. Test. Mater.* **56**, 593–620 (1956).
56. Altenbach, H., Gorash, Y. High-temperature inelastic behavior of the austenitic steel AISI Type 316. In *Advanced Materials Modeling for Structures* 17–30 (2013).
57. Qi-Xun, D., An-Dong, W., Xiao-Nong, C. & Xin-Min, L. Stacking fault energy of cryogenic austenitic steels. *Chin. Phys.* **11**, 596–600 (2002).
58. Punit, K. & Gupta, A. Active learning query strategies for classification, regression, and clustering: A survey. *J. Comput. Sci. Technol.* **35**, 913–945 (2020).
59. Yang, Y. & Loog, M. A variance maximization criterion for active learning. *Pattern Recogn.* **78**, 358–370 (2018).
60. Ren, P. & Xiao, Y. A survey of deep active learning. *ACM Comput. Surv. CSUR* **54**, 1–40 (2021).
61. Pedregosa, F. & Varoquaux, G. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).

62. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **32** (2019).
63. Bingham, E. *et al.* Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.* **20**, 973–978 (2019).

### Acknowledgements

This work was supported by the University of Idaho—Center for Advanced Energy Study (CAES) Seed Funding FY23 Grant. This work was also supported through the INL Laboratory Directed Research & Development (LRD) Program under DOE Idaho Operations Office Contract DE-AC07-05ID14517 (project tracking number 23A1070-069FP). Accordingly, the U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript or allow others to do so, for U.S. Government purposes.

### Author contributions

A.V., Y.W., T.Y., and M.X. developed the concept for the study and designed the experiments. L.L. and J.C. conducted the uncertainty quantification experiments. L.L. conducted the active learning experiments. J.C. conducted the physics-informed neural networks experiments. L.L., A.V., Y.W., T.Y., and M.X. analyzed the results. All authors reviewed the manuscript.

### Competing interests

The authors declare no competing interests.

### Additional information

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1038/s41598-024-61189-x>.

**Correspondence** and requests for materials should be addressed to A.V.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024