

BAYESIAN NEURAL NETWORKS VERSUS DEEP ENSEMBLES FOR UNCERTAINTY QUANTIFICATION IN MACHINE LEARNING INTERATOMIC POTENTIALS

 **Riccardo Farris***

Departament de Ciència de Materials
i Química Física & Institut de Química
Teòrica i Computacional (IQTUB)
Universitat de Barcelona
Barcelona, Spain

 **Emanuele Telari**

Departament de Ciència de Materials
i Química Física & Institut de Química
Teòrica i Computacional (IQTUB)
Universitat de Barcelona
Barcelona, Spain

 **Nongnuch Artrith**

Debye Institute for Nanomaterials Science
Utrecht University
Utrecht, Netherlands

 **Konstantin M. Neyman**

Departament de Ciència de Materials
i Química Física & Institut de Química
Teòrica i Computacional (IQTUB)
Universitat de Barcelona
Barcelona, Spain
ICREA (Institutio Catalana de Recerca i Estudis Avançats)
Pg. Lluís Companys 23
08010 Barcelona, Spain

 **Albert Bruix†**

Departament de Ciència de Materials
i Química Física & Institut de Química
Teòrica i Computacional (IQTUB)
Universitat de Barcelona
Barcelona, Spain

September 24, 2025

ABSTRACT

Neural-network-based machine learning interatomic potentials have emerged as powerful tools for predicting atomic energies and forces, enabling accurate and efficient simulations in atomistic modeling. A key limitation of traditional deep learning approaches, however, is their inability to provide reliable estimates of predictive uncertainty. Such uncertainty quantification is critical for assessing model reliability, especially in materials science, where often the model is applied on out-of-distribution data. Different strategies have been proposed to address this challenge, with deep ensembles and Bayesian neural networks being among the most widely used. In this work, we introduce an implementation of Bayesian neural networks with variational inference in the `ænet`-PyTorch framework. To evaluate their applicability to machine learning interatomic potentials, we systematically compare the performance of variational BNNs and deep ensembles on a dataset of 7,815 TiO_2 structures. The models are trained on both the full dataset and a subset to assess how variations in data representation influence predictive accuracy and uncertainty estimation. This

*Corresponding author: rfarris@ub.edu

†Corresponding author: abruix@ub.edu

analysis provides insights into the strengths and limitations of each approach, offering practical guidance for the development of uncertainty-aware machine learning interatomic potentials.

Keywords Bayesian Neural Networks · Variational Inference · Deep Ensembles · Uncertainty Quantification · Machine Learning Interatomic Potentials

1 Introduction

In recent years, machine learning interatomic potentials (MLIPs) have emerged as powerful tools to overcome the prohibitive computational cost of *ab initio* methods such as density functional theory (DFT). The most promising MLIPs leverage deep learning architectures, such as feed-forward neural networks, and more recently, graph neural networks, to learn the mapping between atomic configurations and their corresponding energies and forces from a limited set of reference calculations [1, 2, 3, 4]. These models have proven capable of achieving superior accuracy and transferability, even when trained on relatively limited datasets. This efficiency dramatically expands the accessible length and time scales, enabling the study of larger and more complex systems that would otherwise be infeasible with traditional first-principles approaches [5, 6].

A critical aspect in the development of such MLIPs is the generation of high-quality datasets for training purposes, due to the high computational cost of first-principles calculations, used as the reference level of theory [7, 8]. This severely limits the size and diversity of training datasets, posing a major bottleneck in the construction of robust and transferable MLIPs. As a result, MLIP models need to be designed not only for accuracy but also for data efficiency, in order to minimize the need for exhaustive DFT sampling while still maintaining predictive reliability across diverse atomic configurations. This calls for finding ways to reliably quantify the uncertainty associated with their predictions, preventing misleading results in simulations, and guiding the selection of informative data points for model refinement in active learning scenarios [9, 10, 11, 12, 13, 14, 15].

Standard deep learning architectures are inherently deterministic and do not provide any measure of prediction uncertainty. As a result, the deployment of deep learning-based MLIPs in uncertainty-sensitive applications requires the incorporation of specialized methods for its quantification [16, 17, 18, 19]. Among the various approaches proposed, deep ensembles (DE) [20] have emerged as one of the most widely adopted techniques [21, 22, 23, 24], due to their simplicity and straightforward application. In this framework, multiple neural networks are independently trained on the same dataset using different initializations, and the variance across their predictions is used as a proxy for model uncertainty [21, 25, 22, 11, 26]. Although DE are effective in practice and relatively simple to implement at inference time, they sometimes lead to overconfident predictions in regions of configuration space that are poorly represented in the training data [27].

In addition to DE, a specialized class of neural networks has emerged to intrinsically offer a measure of their predictions uncertainties by including the Bayesian formalism into the network architecture, the so-called Bayesian Neural Networks (BNNs) [28]. BNNs are grounded in the Bayesian probability framework, which treats model parameters, i.e. weights and biases, as random variables with prior distributions [29]. By incorporating Bayes theorem, BNNs place a posterior distribution over the model parameters conditioned on the training data. Predictions are then obtained by sampling from the resulting predictive distribution, which propagates parameter uncertainty and naturally captures uncertainty in the model’s outputs. Given a dataset \mathbf{X}, \mathbf{Y} where \mathbf{X} is the descriptor matrix and \mathbf{Y} are the reference targets and the model parameters \mathbf{w} , the posterior distribution is computed using Bayes’ theorem:

$$p(\mathbf{w} | \mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y} | \mathbf{X}, \mathbf{w}) p(\mathbf{w})}{p(\mathbf{X} | \mathbf{Y})}, \quad (1)$$

where $p(\mathbf{w})$ is the prior distribution over the parameters in the parameter space Ω , $p(\mathbf{Y} | \mathbf{X}, \mathbf{w})$ is the likelihood of the data given the parameters \mathbf{w} , and $p(\mathbf{X} | \mathbf{Y})$ is the marginal likelihood or evidence.

Based on this posterior, the predictive distribution for a new input \mathbf{x}^* is given by:

$$p(\mathbf{y}^* | \mathbf{x}^*, \mathbf{X}, \mathbf{Y}) = \int_{\Omega} p(\mathbf{y}^* | \mathbf{x}^*, \mathbf{w}) p(\mathbf{w} | \mathbf{X}, \mathbf{Y}) d\mathbf{w} \quad (2)$$

where \mathbf{y}^* is the predicted output, the energy in the case of MLIPs, corresponding to the input features, or descriptor, \mathbf{x}^* . The key challenge in this formulation lies in the intractability of the integral in Equation 2, which arises from the high dimensionality and nonlinearity of the parameter space. As a result, approximate inference techniques must be employed to estimate such distribution. A traditional approach to this problem involves sampling from the posterior

using Markov Chain Monte Carlo (MCMC) methods, including more efficient variants such as Hamiltonian Monte Carlo (HMC) [30]. However, MCMC-based techniques are computationally expensive and often impractical for high-dimensional deep learning models. As a result, variational inference has become a more widely adopted alternative in Bayesian deep learning [31]. Variational inference approximates the true posterior by optimizing over a family of tractable, parameterized distributions, thus enabling scalable and efficient inference at the cost of introducing a variational approximation error [29].

Nonetheless, inferring the posterior distribution in BNNs remains significantly more complex and computationally demanding than training standard deterministic neural networks. This added complexity constitutes a major practical limitation of BNNs, especially when applied to high-dimensional systems as commonly encountered in materials simulations.

In the present work, we explore the applicability and the performance of different techniques to measure the uncertainty of deep learning-based MLIPs, in particular deep ensembles and Bayesian neural networks trained with variational inference, which we will refer to as variational Bayesian neural networks (VBNN). Starting from the `ænet` library [32, 33], we propose a VBNN implementation for MLIPs based on the work of Basora et al. [34], available at <https://github.com/farrisric/bayesaenet>. We employed the Pyro [35] probabilistic programming framework, designed for Bayesian inference, along with the TyXe [36] library, which provides tools for converting standard neural networks into Bayesian neural networks. We then show a detailed comparison between the predictions and uncertainties produced by DE and VBNNs trained with different approaches. The MLIPs have been tested on a dataset comprising 7,815 structures representing various phases of titanium dioxide (TiO₂) [32]. To assess how variations in data representation affect predictive accuracy and uncertainty estimation of the different models, we performed the trainings on both the full dataset and a reduced subset. In Sec. 2 is offered the theoretical background of the different techniques used for uncertainty quantification, i.e. DE and VBNN. Sec. 3 outlines the computational framework developed for the present study, describing the dataset, the networks architectures and hyperparameters, and introducing the evaluation metrics used to assess both predictive accuracy and the quality of uncertainty quantification. Finally, in Sec. 4 the training results and the detailed comparison between the different approaches are reported.

2 Uncertainty quantification in neural networks

2.1 Deep Ensembles

DE [20] consist of M independently trained neural networks $\{f_{\theta_1}, f_{\theta_2}, \dots, f_{\theta_M}\}$, each parametrized by a different random initialization and trained on the same dataset. Given an input atomic configuration x , each model predicts an energy $E_i = f_{\theta_i}(x)$. The ensemble mean prediction is given by:

$$\bar{E} = \frac{1}{M} \sum_{i=1}^M E_i, \quad (3)$$

and the predictive uncertainty is quantified by the sample standard deviation of the ensemble predictions:

$$\sigma_E = \sqrt{\frac{1}{M-1} \sum_{i=1}^M (E_i - \bar{E})^2}. \quad (4)$$

While DEs are attractive because of their ease of implementation, their main drawback is the absence of a rigorous probabilistic foundation: the ensemble variance provides a practical but heuristic measure of uncertainty, rather than one directly derived from a well-defined Bayesian framework. As a result, while DEs often yield reliable estimates in practice, their theoretical justification remains limited compared to fully Bayesian approaches [29, 37].

2.2 Variational Bayesian Neural Networks

Unlike DE, BNNs explicitly model uncertainty in their architecture, by treating weights as random variables with a prior distribution $p(\mathbf{w})$. Given a collection of data \mathcal{D} , the goal is to compute the posterior $p(\mathbf{w}|\mathcal{D})$, which captures plausible weight configurations conditioned on observations. This yields a posterior predictive distribution that marginalizes over all possible weight configurations. However, exact Bayesian inference in deep neural networks is computationally intractable, motivating the use of approximate inference techniques.

Variational Inference [38] approximates the true posterior $p(\mathbf{w}|\mathcal{D})$ over the weights \mathbf{w} , given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, with a variational distribution (also known as guide) $q_\theta(\mathbf{w})$ parametrized by θ . The values of θ are then learned such that the variational distribution $q_\theta(\mathbf{w})$ is as close as possible to the true posterior $p(\mathbf{w}|\mathcal{D})$ [29]. The goal is then to minimize the Kullback-Leibler (KL) divergence [39], which is a measure of similarity of two distributions, between the variational approximation and the true posterior:

$$D_{KL}(q_\theta(\mathbf{w})||p(\mathbf{w}|\mathcal{D})) = \int_{\mathbf{w}} q_\theta(\mathbf{w}) \log \left(\frac{q_\theta(\mathbf{w})}{p(\mathbf{w}|\mathcal{D})} \right) d\mathbf{w} \quad (5)$$

Nonetheless, in order to obtain D_{KL} one still needs to compute $p(\mathbf{w}|\mathcal{D})$. In order to overcome this issue the *evidence lower bound* (ELBO) is used as a loss during the training, defined as:

$$\text{ELBO} = \int_{\mathbf{w}} q_\theta(\mathbf{w}) \log \left(\frac{p(\mathbf{w}, \mathcal{D})}{q_\theta(\mathbf{w})} \right) d\mathbf{w} = \log(p(\mathcal{D})) - D_{KL}(q_\theta(\mathbf{w})||p(\mathbf{w}|\mathcal{D})) \quad (6)$$

Because $\log(p(\mathcal{D}))$ (the log marginal likelihood) is constant with respect to θ , minimizing D_{KL} is equivalent to maximizing the ELBO. The stochastic gradient descent method used to optimize the ELBO is the stochastic variational inference (SVI) [40]

To make variational inference tractable in high-dimensional spaces such as those of deep neural networks, a common simplifying assumption to model the variational distribution $q_\theta(\mathbf{w})$ is the *mean-field approximation*. This assumes that the variational distribution factorizes over individual weights:

$$q_\theta(\mathbf{w}) = \prod_i q_{\theta_i}(w_i) \quad (7)$$

This leads to a fully factorized Gaussian posterior where each weight w_i is modeled with its own mean and variance. While this ignores correlations between weights, treating them as independent variables, it greatly reduces computational complexity and is widely used in practice.

While mean-field approximation offer a relatively simple representation of the variational distribution, optimizing the ELBO with backpropagation remains computationally unfeasible, since the presence of stochastic parameters make standard backpropagation unable to function correctly through internal nodes [41].

A practical implementation of SVI to neural networks is *Bayes by Backprop* [31]. It implements the mean-field approximation and employs the reparametrization trick [42], expressing weights as:

$$\mathbf{w} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$. This formulation allows gradients to flow through stochastic nodes, providing lower-variance gradient estimates than naive score-function methods and making variational Bayesian inference feasible for neural networks.

However, *Bayes by Backprop* makes convergence slow compared to the usual gradient descent as the ELBO is evaluated via Monte Carlo sampling and typically a small number of samples are used, often just one, making its estimate noisy [29]. For this reason, it is typically combined with techniques to reduce the gradient variance, tailored to Bayesian neural networks, such as *local reparametrization trick* (LRT) [43] and *Flipout* (FO) [44]. LRT samples the pre-activations of each data point individually, rather than using a single weight matrix across the batch. This is particularly effective for factorized Gaussian posteriors over the weights in layers performing linear mappings, such as dense or convolutional layers. FO, on the other hand, introduces pseudo-independent perturbations by sampling a rank-one sign matrix per data point. This enables computationally efficient per-example weight sampling while preserving the unbiasedness of gradient estimates, further reducing gradient variance without sacrificing performance.

Finally, variational inference can be treated also outside of the mean-field approximation. Radial BNN (RAD) [45] introduces radial transformations that induce global correlations among weights, modeled as:

$$\mathbf{w} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \frac{\boldsymbol{\epsilon}}{|\boldsymbol{\epsilon}|} \cdot r$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ and $r = |\tilde{r}|$ for $\tilde{r} \sim \mathcal{N}(0, 1)$. These transformations are supposed to better capture heavy-tailed posteriors, reduce gradient variance, and is expected to scale more favourably with larger datasets due to their greater flexibility and expressivity.

3 Computational Framework

3.1 Dataset

The dataset employed in this study consists of 7,815 structures of various titanium dioxide (TiO_2) phases, originally developed to test the \ae net framework [32]. This dataset, which includes different bulk phases of TiO_2 , was also used to benchmark the \ae net-PyTorch implementation. The reference energies were obtained from DFT calculations using the Perdew-Burke-Ernzerhof (PBE) exchange–correlation functional.

The dataset was partitioned into training (80%), validation (10%), and test (10%) subsets. The validation set was used for hyperparameter optimization.

To evaluate the performance and calibration of uncertainty estimates across different data regimes, we conducted experiments using both the full training and validation set (6330 and 704) and a reduced subset comprising 20% of the joint training and validation set (1265, 141), maintaining the test set constant (781 structures). This setup enabled us to assess and compare model behaviour and uncertainty predictions in both high-data and low-data regimes consistently. This ensures a case in which the training data represent a complete structural picture of the system (high-data), and a case in which the system is under-represented (low-data).

3.2 Model Architectures

3.2.1 Neural Networks

All neural networks used in this study shared the same architecture: two hidden layers with 15 units each and hyperbolic tangent (\tanh) activation functions. This lightweight configuration results in approximately 2,700 trainable parameters, providing a balance between expressivity and computational efficiency. The models were implemented using PyTorch, with training managed via the PyTorch Lightning framework [46] to ensure modularity and reproducibility.

3.2.2 Variational Inference

For variational inference, we tested two types of variational guides. The first is based on mean-field approximation, AutoNormal guide as implemented in TyXe [36], which samples all unobserved sites from a diagonal Gaussian distribution, thus approximating the posterior with a fully factorized normal distribution. We used *Bayes by Backprop* to optimize the network using two different technique for gradient variance reduction: LRT and FO. The other type of guide tested is the Radial guide (RAD), which performs a radial transformation over the weights.

3.3 Model Training

All models were trained under both high-data and low-data regimes, based on the full TiO_2 dataset. In the high-data regime, the dataset was partitioned into an 80% training set, a 10% validation set for hyperparameter optimization, and a 10% test set for performance evaluation. In the low-data regime, the test set remained fixed, while only 20% of the entire dataset was used for training and validation (with an 80/20 split within that subset).

The Deep Ensemble (DE) model consisted of 10 independently trained neural networks, each optimized using the root mean squared error (RMSE) loss. In contrast, the Bayesian Neural Networks (BNNs) were trained using variational inference by maximizing the evidence lower bound (ELBO).

To ensure robustness and mitigate the influence of random initialization, each model configuration was trained five times using different random seeds. All models were trained for a maximum of 100,000 epochs with early stopping based on validation loss. A patience of 100 epochs was applied for both DE and BNNs to prevent overfitting and ensure convergence.

3.4 Evaluation Metrics

To assess the accuracy and reliability of the predictive models, we employ several complementary evaluation metrics that quantify both point prediction performance and the quality of uncertainty quantification (UQ).

For model accuracy, we report the Mean Absolute Error (MAE), the Root Mean Squared Error (RMSE), and the Negative Log-Likelihood (NLL). These metrics provide insights into different aspects of prediction error: MAE captures the average deviation regardless of direction, RMSE emphasizes larger errors and is sensitive to outliers, and NLL offers a combined score that evaluates both the accuracy and the quality of the uncertainty estimates [47]. The MAE, RMSE and NLL are defined as:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2, \quad (8)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2}, \quad (9)$$

$$\text{NLL} = \frac{1}{N} \sum_{i=1}^N \left[\frac{(y_i - \hat{y}_i)^2}{2\sigma_i^2} + \frac{1}{2} \log(2\pi\sigma_i^2) \right], \quad (10)$$

where \hat{y}_i and y_i denote the predicted and true values, respectively, and σ_i^2 is the predictive variance for data point i .

To evaluate the quality of uncertainty quantification (UQ), we employ calibration metrics, which assess how well predicted confidence intervals correspond to actual outcome frequencies. In particular, we use calibration curves, which plot the empirical frequency of true values falling within a given predicted confidence level. Ideally, a model’s 90% confidence intervals should contain the true value 90% of the time. Calibration curves thus provide a direct visual assessment of how well uncertainty estimates match empirical behavior [48].

We also quantify miscalibration using the Root Mean Squared Calibration Error (RMSCE), which measures the area between the calibration curve and the diagonal representing perfect calibration. Lower RMSCE values indicate better uncertainty estimates.

Figure 1 illustrates three prototypical cases of model calibration:

- Left: An overconfident model predicts uncertainty intervals that are too narrow, failing to encompass the true values as often as expected. This results in a calibration curve that lies below the diagonal and a substantial miscalibration area.
- Center: An underconfident model produces overly broad intervals, capturing the true values more frequently than required. The corresponding calibration curve lies above the diagonal, reflecting an overly cautious model.
- Right: A well-calibrated model yields prediction intervals that closely match empirical coverage across all confidence levels. Its calibration curve follows the diagonal closely, and the miscalibration area is minimal.

These examples underscore the importance of both accurate and well-calibrated uncertainty estimates, particularly when predictive models are applied to tasks where trust and interpretability are critical.

However, calibration alone does not fully characterize the informativeness of uncertainty estimates. A model can be well-calibrated while still producing overly broad predictions. To complement calibration, we therefore compute the Sharpness, which measures the average predictive uncertainty:

$$\text{Sharpness} = \frac{1}{N} \sum_{i=1}^N \sigma_i. \quad (11)$$

Here, σ_i is the predicted standard deviation for data point i . Lower sharpness indicates more confident (narrower) predictions. In an ideal model, low sharpness is achieved without compromising calibration, yielding both informative and trustworthy uncertainty estimates.

In the context of a hypothetical active learning scenario, we introduce two custom metrics to assess the alignment between predicted uncertainties and actual prediction errors. The first is the coefficient of determination (R^2) between predicted uncertainty (standard deviation σ) and the model’s absolute error (MAE). This metric quantifies the extent to which uncertainty estimates can explain the variance in predictive errors via a linear regression model. A higher R^2 value indicates a stronger correlation, which is desirable in active learning where uncertainty is used to guide data acquisition.

The second metric, which we name the **Overlap Score**, is designed to capture the consistency between high-error and high-uncertainty regions—areas of particular interest for active learning algorithms, which typically prioritize samples where model confidence is low and prediction error is high. By quantifying the alignment between these regions, the overlap score provides a practical diagnostic for evaluating the suitability of uncertainty estimates in guiding data acquisition. Both the predicted uncertainties and the corresponding absolute errors are discretized into quartiles. The Overlap Score is defined as the percentage of high-uncertainty predictions that also lie in the high-error region. This

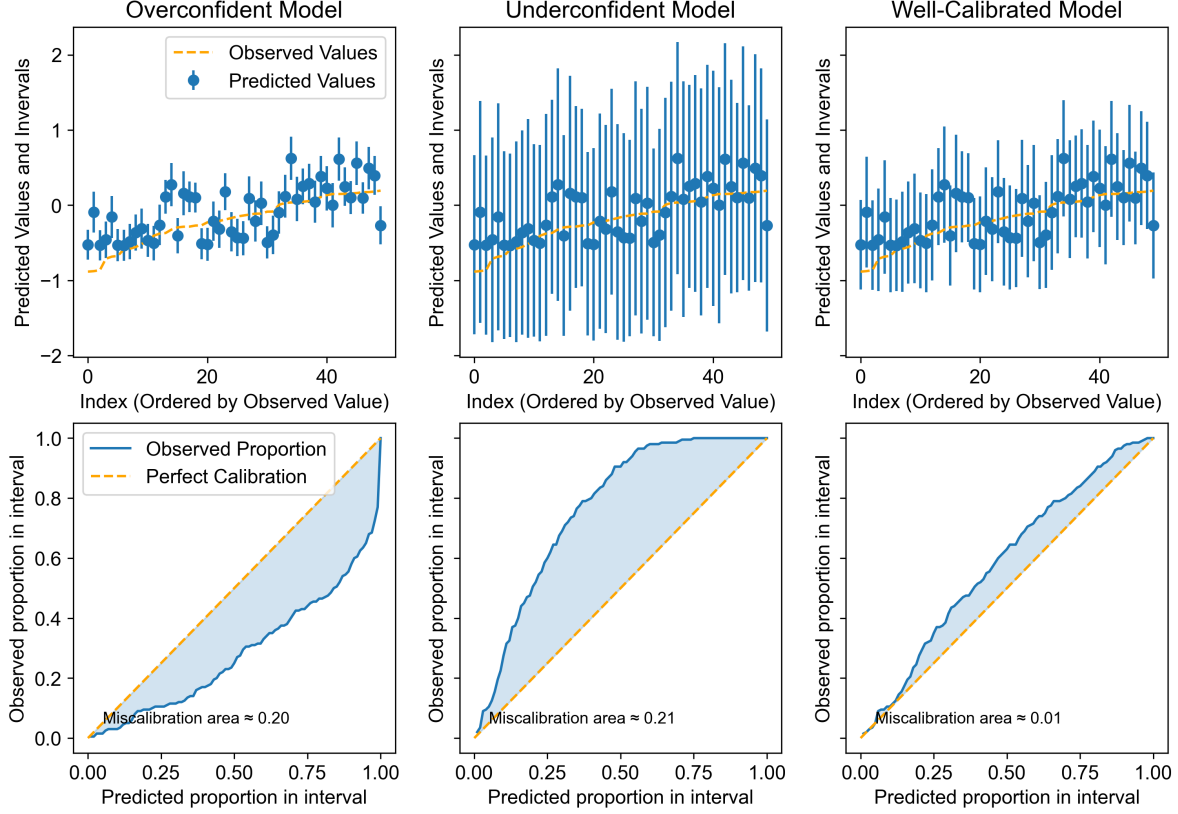


Figure 1: Illustration of uncertainty calibration behavior in three types of models. Top: predicted values ordered by observed values. Bottom: calibration curves showing observed vs. predicted proportions of values within confidence intervals. Left: overconfident model; center: underconfident model; right: well-calibrated model.

answers the question: *Among the data points the model is least certain about, how many are actually among the most inaccurate?* This makes the metric particularly suitable for evaluating the usefulness of uncertainty estimates in active learning or reliability assessment.

Formally, let the true target values be $\mathbf{y}_{\text{true}} = \{y_1^{\text{true}}, \dots, y_n^{\text{true}}\}$, the predicted values $\mathbf{y}_{\text{pred}} = \{y_1^{\text{pred}}, \dots, y_n^{\text{pred}}\}$, and the predicted standard deviations (uncertainties) $\boldsymbol{\sigma} = \{\sigma_1, \dots, \sigma_n\}$. Define the absolute error for each sample as

$$e_i = |y_i^{\text{true}} - y_i^{\text{pred}}|. \quad (12)$$

Let Q_3^{error} and $Q_3^{\text{uncertainty}}$ be the third quartile (75th percentile) of the absolute error and uncertainty distributions, respectively. Then, define the indicator functions

$$H_i^{\text{error}} = \begin{cases} 1 & \text{if } e_i > Q_3^{\text{error}} \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad H_i^{\text{uncertainty}} = \begin{cases} 1 & \text{if } \sigma_i > Q_3^{\text{uncertainty}} \\ 0 & \text{otherwise.} \end{cases}$$

The overlap score is then given by

$$\text{Overlap Score} = 100 \times \frac{\sum_{i=1}^n H_i^{\text{error}} \cdot H_i^{\text{uncertainty}}}{\sum_{i=1}^n H_i^{\text{uncertainty}}}. \quad (13)$$

An illustration of the overlap metric is provided in Figure 2, where each point represents a single prediction plotted by its associated uncertainty (x-axis) and absolute error (y-axis). The red points in the upper-right quadrant highlight cases of simultaneously high uncertainty and high error, whose density reflects the degree of overlap.

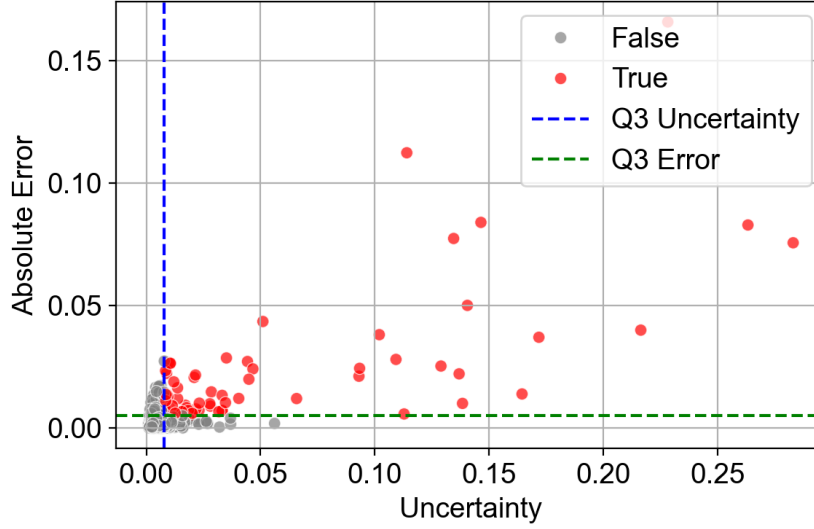


Figure 2: Illustration of the overlap metric. Each dot is a prediction with its uncertainty (x-axis) and absolute error (y-axis). Red points represent high-uncertainty, high-error cases in the upper right quadrant.

3.5 Hyperparameter Optimization

Hyperparameter optimization was performed using the Optuna framework [49], with 60 trials trained for 500 epochs conducted for each model across both dataset sizes. The hyperparameters optimized for the Bayesian models were:

- **Learning rate** [10^{-5} , 10^{-3}]: Determines the step size used by the optimizer during gradient updates.
- **Batch Size** [32, 64, 128, 256]: Specifies the number of training samples used to compute a single optimization step, i.e., one forward and backward pass through the model followed by a parameter update. Larger batches produce more stable gradient estimates but may reduce generalization; smaller batches introduce noise, which can help escape local minima.
- **Number of MC samples** [1, 2]: The number of Monte Carlo samples drawn from the variational posterior q_θ to estimate the ELBO. A higher number yields more accurate gradient estimates but incurs greater computational cost.
- **Gaussian prior scale** [0.1, 1.5]: Controls the variance of the prior distribution over weights. Larger values impose stronger regularization, encouraging simpler models.
- **Gaussian q_θ scale** [10^{-4} , 0.1]: Sets the initial variance of the variational posterior. Smaller values produce narrower posteriors and more confident predictions.
- **Likelihood variance** [0.1–2.0]: Variance of the assumed homoscedastic Gaussian likelihood, capturing aleatoric uncertainty in the data.

The optimal hyperparameter configurations identified for each model and data regime are reported in Table 1 and Table 2. Initially, optimization was performed using the ELBO on the validation set as the objective. However, we observed that this frequently led to highly negative ELBO values and unstable uncertainty estimates. Switching the optimization objective to the mean squared error (MSE) on the validation set resulted in significantly improved stability and more reliable uncertainty quantification for the VBNNs.

The optimal hyperparameters selected via Optuna (Tables 2 and 1) reveal several consistent trends and informative differences across both model types and data regimes. First, the number of Monte Carlo samples was fixed at 2 across all models and data regimes. This reflects a compromise between stability in ELBO estimation and computational efficiency, as using more samples would reduce the variance of gradient estimates but significantly increase training time.

In terms of the learning rate, we observe that in the low-data regime (20%), all models favour higher values compared to the high-data regime (100%). This behaviour is consistent with the reduced number of training samples, where a

Table 1: Best hyperparameters selected via Optuna for each model trained on 20% of the dataset.

Hyperparameter	FO 20%	LRT 20%	RAD 20%
Learning Rate	0.000325	0.000540	0.000137
Batch Size	64	64	32
MC Samples (Train)	2	2	2
Prior Scale	0.175	0.358	0.115
q_θ Scale	0.001832	0.001246	0.000172
Likelihood variance	0.260	0.282	0.793

Table 2: Best hyperparameters selected via Optuna for each model trained on 100% of the dataset.

Hyperparameter	FO 100%	LRT 100%	RAD 100%
Learning Rate	0.000124	0.000049	0.000536
Batch Size	256	64	64
MC Samples (Train)	2	2	2
Prior Scale	0.206	0.209	0.108
q_θ Scale	0.000605	0.000227	0.000800
Likelihood variance	0.893	0.132	0.294

slightly more aggressive learning rate can accelerate convergence without compromising generalization, provided that the variance of updates is adequately controlled.

The prior scale and variational posterior scale (q_θ) values indicate differing levels of regularization across regimes. In the high-data regime, the models tend to prefer tighter posteriors (e.g., lower q_θ scales in LRT and FO), suggesting that the larger training set supports sharper and more confident posterior distributions. In contrast, the posterior scale increases in the low-data regime for FO and LRT, reflecting the model’s need to retain more uncertainty due to limited training data. Interestingly, RAD remains relatively conservative in both regimes, with moderate prior and posterior scales, potentially due to its heavier-tailed variational family.

The observation scale (i.e., the variance of the likelihood) also varies meaningfully. In the low-data regime, the observation noise is generally higher than in the high-data case, suggesting that the models compensate for reduced data availability by attributing more of the residual variance to aleatoric uncertainty. The exception is FO in the high-data regime, which shows a surprisingly large observation scale (0.893), which may indicate that the model is not adequately capturing the underlying structure (underfitting) and is over-attributing residual error to data noise rather than to model uncertainty.

Finally, batch size choices reflect a trade-off between noise in gradient estimates and regularization effects. Larger batch sizes were selected for FO (both regimes), while RAD and LRT favour smaller batches.

4 Results and Discussion

4.1 Predictive Accuracy

The predictive accuracy of each model is summarized in Figure 3, which reports the results for both the high-data and low-data regimes. The evaluation metrics considered are MAE, RMSE, and NLL, as introduced in Section 3.4. Each bar in the figure represents the distribution of the metric values obtained from five independently trained models using different random seeds. This allows us to assess the robustness and variability of each method under repeated training.

Table 3 reports the predictive performance of each method in the high-data regime (100% of the dataset). All models achieved relatively low MAE and RMSE values, confirming the effectiveness of both ensemble and variational strategies when sufficient data is available.

DE achieved the best performance across all metrics, with MAE of 0.005 and RMSE of 0.012, significantly outperforming the Bayesian counterparts. FO and LRT followed closely, with FO showing MAE of 0.017 and RMSE of 0.028, while LRT achieved slightly better MAE at 0.014 but slightly worse RMSE at 0.029. RAD lagged behind with the highest errors: MAE of 0.019 and RMSE of 0.041.

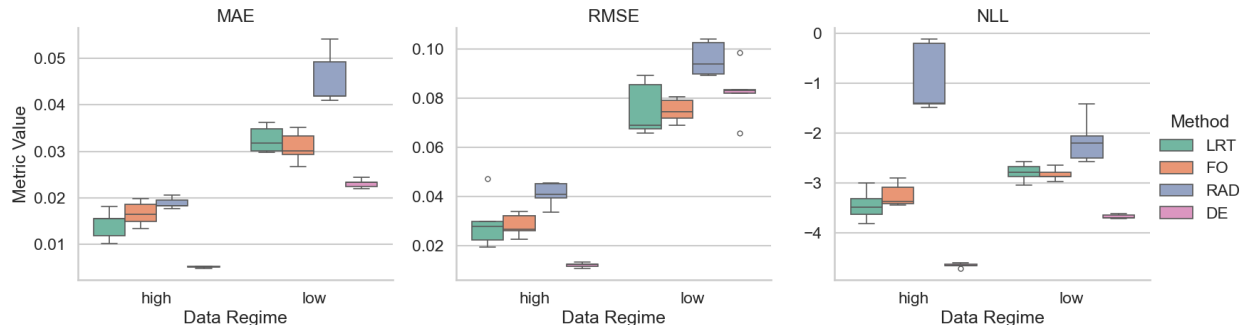


Figure 3: Predictive performance metrics (MAE, RMSE, NLL) for all models across high-data (100% of the dataset) and low-data (20% of the dataset) regimes. All metrics are reported in eV/atom. Boxplots summarize the distribution of scores over five independent training runs: the boxes show the interquartile range (25th to 75th percentile), the horizontal line within each box marks the median, and the whiskers extend to the most extreme non-outlier points. Coloured points outside the whiskers represent outliers (fliers), indicating training runs with unusually high or low scores. Colour coding: LRT (green), FO (orange), RAD (blue), and DE (pink).

The negative log-likelihood (NLL) values further underscore these differences. DE obtained the most negative NLL at -4.65, indicating not only high predictive accuracy but also well-calibrated uncertainty estimates. FO (-3.24) and LRT (-3.44) also provided reasonable calibration. In contrast, RAD yielded a markedly worse NLL of -0.93.

Table 3: Average MAE, RMSE, and NLL for each model trained on 100% of the dataset. All values are reported in eV/atom, with uncertainty indicating the standard deviation over five independent runs.

Method	MAE	RMSE	NLL
DE	0.005 \pm 0.001	0.012 \pm 0.001	-4.65 \pm 0.04
FO	0.017 \pm 0.003	0.028 \pm 0.005	-3.24 \pm 0.24
LRT	0.014 \pm 0.003	0.029 \pm 0.011	-3.44 \pm 0.31
RAD	0.019 \pm 0.001	0.041 \pm 0.005	-0.93 \pm 0.69

Table 4 summarizes the predictive performance of all models in the low-data regime. Despite the reduced training set, DE retained its overall superiority in accuracy, achieving the lowest MAE of 0.023 and RMSE of 0.082. Among the Bayesian models, FO and LRT remained competitive: FO reached an MAE of 0.031 and an RMSE of 0.075, while LRT followed closely with 0.033 and 0.075, respectively. RAD again delivered the weakest results, with the highest errors (MAE = 0.046, RMSE = 0.096) and broader variability.

Interestingly, FO and LRT achieved lower RMSE values than DE in this regime. This suggests that these Bayesian models are more robust to outliers, particularly FO, which also exhibited the smallest RMSE standard deviation among all models.

NLL results corroborate these findings. DE preserved the lowest NLL value (-3.67), reflecting its continued strength in both predictive accuracy and uncertainty calibration. FO and LRT also performed reasonably well, with NLL values of -2.83 and -2.79, respectively. RAD, in contrast, remained the least reliable, with a higher NLL of -2.15 and a significantly larger standard deviation.

Table 4: Average MAE, RMSE, and NLL for each model trained on 20% of the dataset. All values are reported in eV/atom, with the standard deviation over five independent runs.

Method	MAE	RMSE	NLL
DE	0.023 \pm 0.001	0.082 \pm 0.012	-3.67 \pm 0.04
FO	0.031 \pm 0.003	0.075 \pm 0.005	-2.83 \pm 0.13
LRT	0.033 \pm 0.003	0.075 \pm 0.011	-2.79 \pm 0.18
RAD	0.046 \pm 0.006	0.096 \pm 0.007	-2.15 \pm 0.46

In summary, DE consistently achieved the lowest values across all evaluated metrics—including MAE, RMSE, and especially NLL, highlighting its superior predictive accuracy and more reliable uncertainty quantification compared to the Bayesian alternatives.

Among the Bayesian approaches, both FO and LRT yielded comparable results. Although their predictive accuracy did not match the performance of DE, they remained competitive across metrics. Notably, both methods demonstrated greater robustness to outliers, as indicated by lower RMSE values. This reduced sensitivity to extreme prediction errors suggests that variational strategies such as FO and LRT may offer enhanced generalization capabilities under low-data conditions.

Conversely, RAD performed the worst across both data regimes. Its elevated error metrics and poor NLL values suggest instability during training, likely due to the incompatibility of the RAD guide with low-variance gradient estimators or due to the limited size of the model.

Must be also noticed that Bayesian approaches appear to be more influenced by the initial random initialization respect to DE. This is most likely due to the higher complexity of the training and due to the noisy gradients.

4.2 Uncertainty Quantification

The uncertainty quantification (UQ) performance of each model is summarized in Figure 4, which presents results for both the high-data and low-data regimes. The evaluation relies on four metrics: root mean squared calibration error (RMSCE), sharpness (SHARP), coefficient of determination (R^2), and overlap score, as defined in Section 3.4.

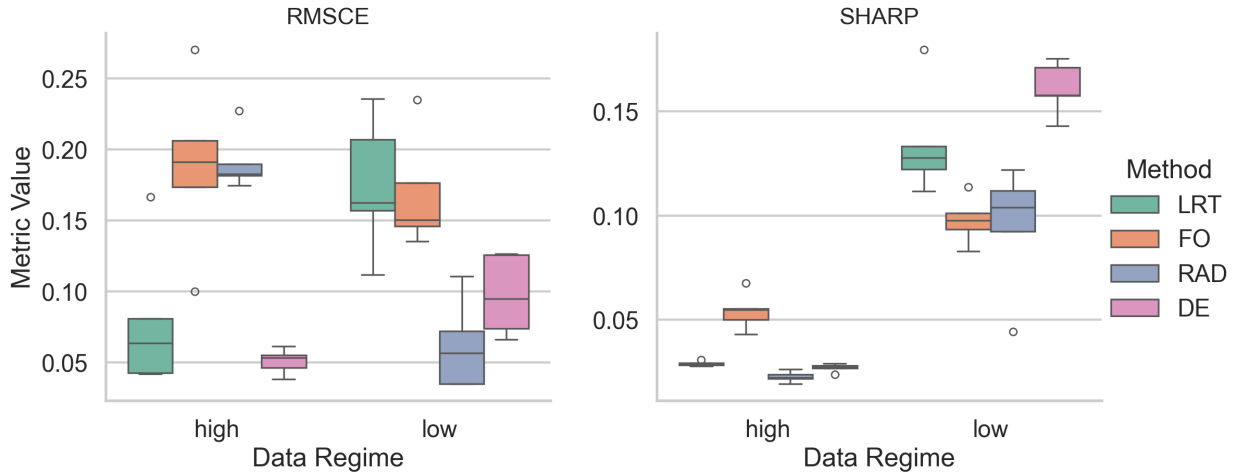


Figure 4: Root mean squared calibration error (RMSCE) and sharpness (SHARP) of the predicted uncertainty distributions across methods and data regimes. Box plots summarize the distribution of scores over five independent training runs: the boxes show the interquartile range (25th to 75th percentile), the horizontal line within each box marks the median, and the whiskers extend to the most extreme non-outlier points. Coloured points outside the whiskers represent outliers (fliers), indicating training runs with unusually high or low scores. Colour coding: LRT (green), FO (orange), RAD (blue), and DE (pink).

Table 5 shows the metrics mean and standard deviation of the 5 models in the high-data regime. DE achieved the best overall calibration with the lowest RMSCE of 0.05, highlighting its ability to produce well-calibrated uncertainty estimates when sufficient training data are available. Among Bayesian approaches, LRT also performed competitively (RMSCE = 0.08), whereas FO and RAD exhibited substantially higher calibration errors (RMSCE = 0.19 and 0.19, respectively).

In terms of sharpness, DE again outperformed the alternatives, exhibiting the lowest value (SHARP = 0.027), indicative of confident and concentrated predictive distributions. RAD produced the best SHARP results (0.022) although its poor calibration undermines its reliability. FO, with the highest SHARP value (0.054), produced broader uncertainty intervals, consistent with a tendency to overestimate predictive uncertainty.

Table 6 shows the metrics mean and standard deviation of the 5 models in the low-data regime. All models exhibited a clean degradation in calibration quality, as indicated by increased RMSCE values. Among them, RAD achieved the best calibration (RMSCE = 0.06) and the sharpest uncertainty estimates (SHARP = 0.09), suggesting it offered the

Table 5: Average RMSCE, SHARP, R^2 and OVERLAP for each model trained on 100% of the dataset. Standard deviation over five independent runs.

Method	RMSCE	SHARP	R^2	OVERLAP
DE	0.05 ± 0.01	0.027 ± 0.002	0.51 ± 0.10	45 ± 4
FO	0.19 ± 0.06	0.054 ± 0.009	0.25 ± 0.06	45 ± 5
LRT	0.08 ± 0.05	0.033 ± 0.001	0.23 ± 0.06	43 ± 5
RAD	0.19 ± 0.02	0.022 ± 0.003	0.33 ± 0.05	44 ± 2

most favourable UQ performance in this setting. DE ranked second in calibration (RMSCE = 0.10) but recorded the highest sharpness value (0.16), reflecting broader and less confident uncertainty estimates. FO and LRT experienced the most pronounced drops in calibration reliability, with RMSCE values of 0.17 and 0.17, respectively. While FO demonstrated relatively sharp predictions (SHARP = 0.10), this was overshadowed by its elevated calibration error, indicating a mismatch between predictive confidence and actual performance.

Table 6: Average RMSCE, SHARP, R^2 and OVERLAP for each model trained on 20% of the dataset. Standard deviation over five independent runs.

Method	RMSCE	SHARP	R^2	OVERLAP
DE	0.10 ± 0.03	0.16 ± 0.01	0.59 ± 0.09	59 ± 4
FO	0.17 ± 0.04	0.10 ± 0.01	0.33 ± 0.15	52 ± 5
LRT	0.17 ± 0.05	0.13 ± 0.03	0.35 ± 0.05	51 ± 4
RAD	0.06 ± 0.03	0.09 ± 0.03	0.30 ± 0.08	46 ± 1

Figure 5 provides a visual assessment of the uncertainty quality for the best-performing models in the low-data and high-data regimes, as selected based on the combined RMSCE and SHARP metrics. In the top row, calibration curves show how closely the predicted confidence intervals match the empirical coverage. A well-calibrated model should align closely with the diagonal dashed line (Ideal).

Under the low-data regime (left), the RAD model demonstrates reasonably good calibration overall; however, it also exhibits one of the highest sharpness values, showing that Table 4 does not fully capture the trade-offs between calibration and confidence. This highlights the importance of jointly considering both metrics when evaluating uncertainty quality. In the high-data regime (right), DE achieves near-ideal calibration, followed by the almost identical LRT.

The lower row of Figure 5 depicts sharpness through violin plots of the predicted standard deviations. Narrower and more concentrated distributions near zero indicate more confident (i.e., sharper) predictions. As expected, all models demonstrate improved sharpness in the high-data regime relative to the low-data regime, with DE producing the sharpest distributions in both cases. LRT closely matches DE in calibration performance but exhibits slightly higher predictive variance. In contrast, RAD again displays strong performance in sharpness but poorer calibration, illustrating a misalignment between prediction confidence and empirical coverage.

We further assessed the quality of uncertainty quantification using the coefficient of determination (R^2) and the overlap score. As shown in Figure 6 and reported in the fourth and fifth columns of Tables 5 and 6, DE consistently achieved the highest R^2 values across both data regimes. In the high-data regime, DE obtained an R^2 of 0.5133 ± 0.1011 , which improved to 0.5882 ± 0.0939 in the low-data setting, indicating a strong correlation between the predicted uncertainty and the actual model error. In contrast, the variational methods demonstrated lower R^2 values in the low-data regime: FO and LRT achieved 0.3268 and 0.3491, respectively, while RAD scored slightly lower at 0.2953. These results highlight the reduced ability of variational approaches to capture the true variability in model error.

The overlap score trends mirrored the R^2 findings. In the high-data regime, all methods achieved comparable overlap scores, with FO marginally outperforming others (45), followed closely by DE (45), RAD (44), and LRT (43). However, under the low-data regime, DE significantly outperformed the other methods, achieving an overlap score of 59. FO and LRT followed with 52 and 51, respectively, while RAD underperformed at 46. These findings emphasize DE’s superior reliability in capturing the true variability and coverage of predictions under both abundant and scarce data conditions.

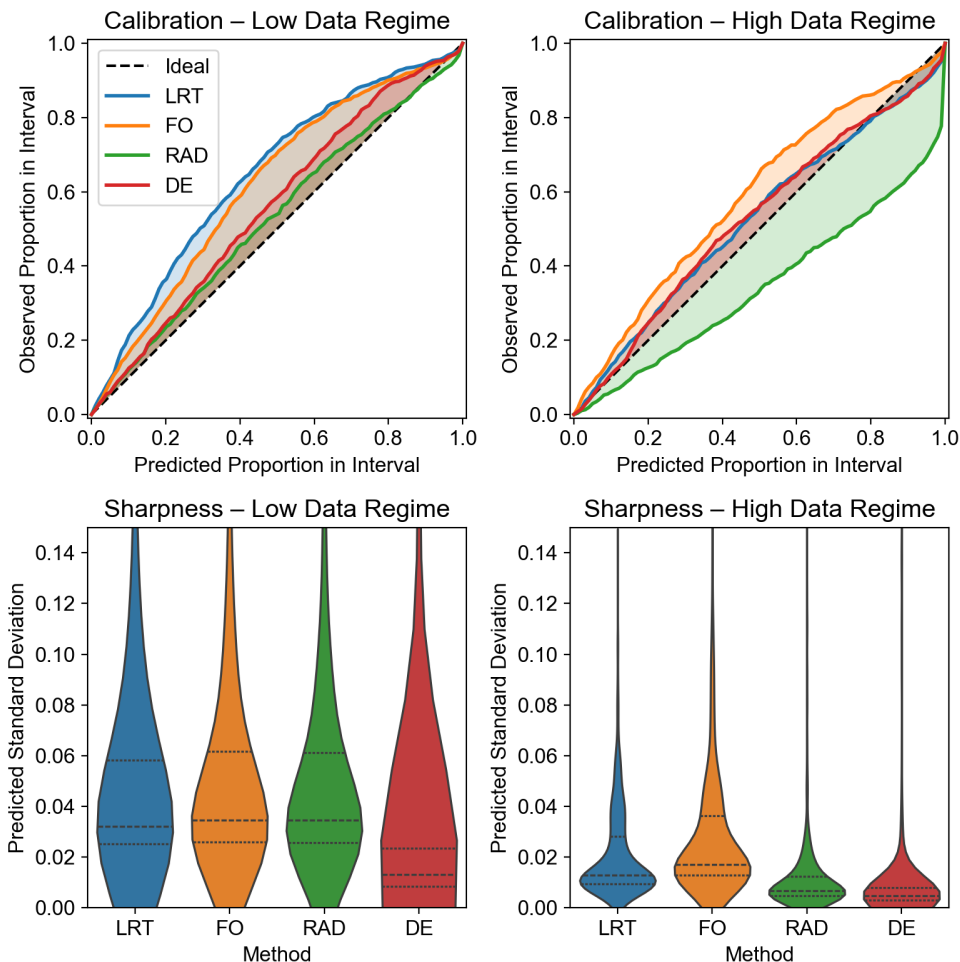


Figure 5: Calibration curves (top) and sharpness distributions (bottom) for the best-performing models in each data regime, selected based on the lowest combined RMSCE and SHARP scores.

4.3 Training and Convergence Time

Table 7 presents a comparative overview of training efficiency across different uncertainty-aware models under both low- and high-data regimes. The deep ensemble (DE) model, formed by aggregating 10 independently trained neural networks, retains low overall wall-clock time despite having a higher total epoch count, highlighting the efficiency of individual NN components. In contrast, Bayesian approaches—LRT, FO, and RAD—require substantially more computational effort, with longer convergence times and greater variability, especially in the low-data setting. This variance reflects the sensitivity of these models to data scarcity and the stochasticity of Bayesian optimization.

It is also worth noting that all models in this study were trained using a single-core CPU, representing a lower-bound baseline in terms of computational resources. Substantial gains in efficiency could be achieved by leveraging GPU acceleration, particularly for Bayesian Neural Networks.

5 Conclusions

In this work, we conducted a systematic comparison of different techniques for uncertainty estimation for neural networks in the context of machine learning interatomic potentials. We tested Deep Ensembles (DE) and three variational Bayesian neural network (VBNN) approaches: Local Reparametrization Trick (LRT), Flipout (FO), and Radial guide (RAD). The evaluation was carried out on a pre-existing TiO_2 dataset with various crystalline phases of

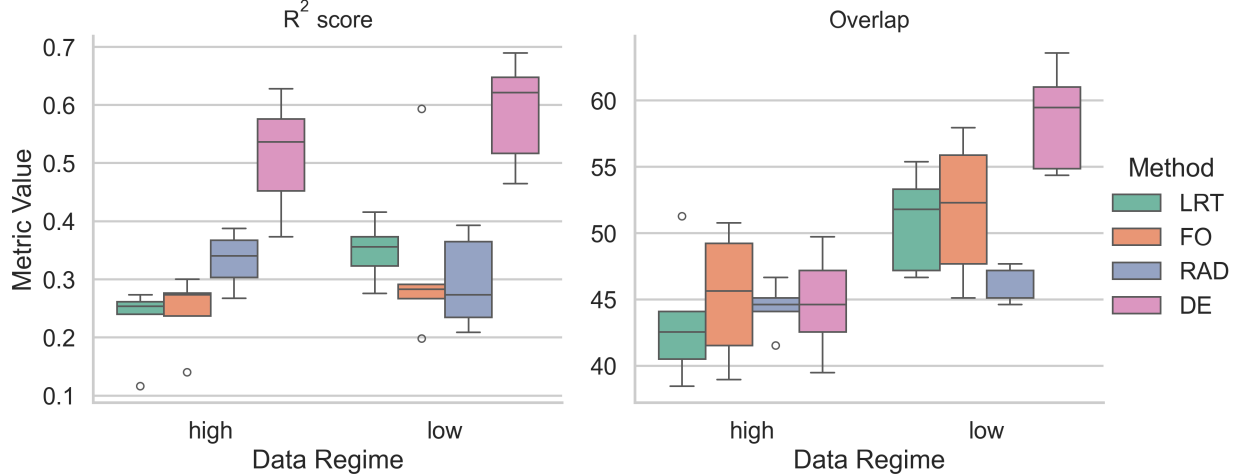


Figure 6: Quality of uncertainty estimation as assessed by coefficient of determination (R^2) and overlap score across models and data regimes. Boxplots summarize the distribution of scores over five independent training runs: the boxes show the interquartile range (25th to 75th percentile), the horizontal line within each box marks the median, and the whiskers extend to the most extreme non-outlier points. Coloured points outside the whiskers represent outliers (fliers), indicating training runs with unusually high or low scores. Colour coding: LRT (green), FO (orange), RAD (blue), and DE (pink).

Table 7: Mean and standard deviation of training epochs and wall-clock time (in minutes) for each model in both the low-data and high-data regimes, computed from multiple runs. Epochs and time for DE are multiplied by 10, having performed 10 individual trainings.

Regime	Model	Epochs to Converge	Time (min)
Low	LRT	8,408 \pm 3,969	93 \pm 42
	FO	29,384 \pm 20,804	320 \pm 223
	RAD	21,755 \pm 15,892	409 \pm 322
	DE	37,600 \pm 8,459	40 \pm 10
High	LRT	59,262 \pm 21,115	3,030 \pm 982
	FO	40,327 \pm 28,709	1,133 \pm 866
	RAD	28,980 \pm 8,237	1,086 \pm 334
	DE	81,980 \pm 4825	350 \pm 22

the oxide [32]. We assessed model performance across both high-data and low-data regimes using a comprehensive suite of metrics designed to quantify both predictive accuracy and the quality of uncertainty quantification.

Our findings showed that, regardless of the more empirical and less-theory-grounded approach, DE consistently demonstrated superior performance across almost all metrics and the two data regimes. DE achieved the highest predictive accuracy (lowest MAE and RMSE), the most calibrated uncertainty estimates (lowest RMSCE and NLL), and the most informative uncertainty quantification, as reflected by the highest R^2 and overlap scores. Notably, in the low-data regime, DE exhibited increased underconfidence, evidenced by its reduced sharpness compared to other models. Nonetheless, it maintained the strongest correlation between predicted errors and predicted uncertainties. Although the non-bayesian framework, it is not new that DE empirically outperforms BNNs. Recent work has shown that DE can be interpreted as performing empirical Bayes inference wherein the prior is learned from the data [37] and they can be interpreted as a multi-modal sampling of the posterior distribution [29].

Among the VBNN approaches, FO and LRT offered a reasonable trade-off between accuracy and UQ. Both methods showed competitive performance, particularly in their robustness to outliers, as demonstrated by lower RMSE variance. However, their calibration and uncertainty correlation deteriorated more significantly than DE under low-data conditions.

These findings suggest that ensemble-based methods, such as DE, offer major advantages for uncertainty-aware modelling, especially due to their lower complexity and computational demand. On the other hand, VBNN offered

overall good and comparable performance, although at a higher computational effort. Moreover, the complexity of the VBNN training task makes them more susceptible to random initialization, suggesting that more than one training should be performed to achieve ideal performance. Nonetheless, such higher complexity, related to their stronger theoretical background still makes them an interesting tool to quantify uncertainty, allowing for a more informed and tunable modelling over the specific dataset.

Acknowledgements

This work was supported by the Spanish/FEDER Ministerio de Ciencia, Innovacion y Universidades [Grant Nos. PID2021-128217NB-I00, MDM-2017-0767, CEX2021-001202-M, PID2022-140120OA-I00, and RYC2021-032281-I (for A.B.)] as well as by the Generalitat de Catalunya [Grant No. and 2021SGR00286]. R. F. thanks the Spanish MICIUN for an FPI PhD grant (MDM-2017-0767-20-2). Computer resources have been partly provided by the Red Española de Supercomputacion. This study was also supported by the European COST Actions CA18234 and CA21101.

Data Availability

The implemented library (along with documentation) are available in the GitHub repository bayesaenet (<https://github.com/farrisric/bayesaenet>).

References

- [1] J. Behler and M. Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.*, 98:146401–146404, 2007.
- [2] Nongnuch Artrith and Alexie M. Kolpak. Understanding the composition and activity of electrocatalytic nanoalloys in aqueous solvents: A combination of DFT and accurate neural network potentials. *Nano Letters*, 14(5):2670–2676, 2014. ISSN 15306992. doi:10.1021/nl5005674.
- [3] Ilyes Batatia, Philipp Benner, Yuan Chiang, Alin M. Elena, Dávid P. Kovács, Janosh Riebesell, Xavier R. Advincula, Mark Asta, Matthew Avaylon, William J. Baldwin, Fabian Berger, Noam Bernstein, Arghya Bhowmik, Samuel M. Blau, Vlad Cărare, James P. Darby, Sandip De, Flaviano Della Pia, Volker L. Deringer, Rokas Elijošius, Zakariya El-Machachi, Fabio Falcioni, Edvin Fako, Andrea C. Ferrari, Annalena Genreith-Schriever, Janine George, Rhys E. A. Goodall, Clare P. Grey, Petr Grigorev, Shuang Han, Will Handley, Hendrik H. Heenen, Kersti Hermansson, Christian Holm, Jad Jaafar, Stephan Hofmann, Konstantin S. Jakob, Hyunwook Jung, Venkat Kapil, Aaron D. Kaplan, Nima Karimitari, James R. Kermode, Namu Kroupa, Jolla Kullgren, Matthew C. Kuner, Domantas Kuryla, Guoda Liepuoniute, Johannes T. Margraf, Ioan-Bogdan Magdău, Angelos Michaelides, J. Harry Moore, Aakash A. Naik, Samuel P. Niblett, Sam Walton Norwood, Niamh O’Neill, Christoph Ortner, Kristin A. Persson, Karsten Reuter, Andrew S. Rosen, Lars L. Schaaf, Christoph Schran, Benjamin X. Shi, Eric Sivonxay, Tamás K. Stenczel, Viktor Svahn, Christopher Sutton, Thomas D. Swinburne, Jules Tilly, Cas van der Oord, Eszter Varga-Umbrich, Tejs Vegge, Martin Vondrák, Yangshuai Wang, William C. Witt, Fabian Zills, and Gábor Csányi. A foundation model for atomistic materials chemistry, 2024. URL <https://arxiv.org/abs/2401.00096>.
- [4] Volker L. Deringer, Miguel A. Caro, and Gábor Csányi. Machine learning interatomic potentials as emerging tools for materials science. *Advanced Materials*, 31(46):1902765, 2019. doi:<https://doi.org/10.1002/adma.201902765>. URL <https://advanced.onlinelibrary.wiley.com/doi/abs/10.1002/adma.201902765>.
- [5] Jörg Behler. Perspective: Machine learning potentials for atomistic simulations. *The Journal of chemical physics*, 145(17), 2016.
- [6] Yunxing Zuo, Chi Chen, Xiangguo Li, Zhi Deng, Yiming Chen, Jörg Behler, Gábor Csányi, Alexander V Shapeev, Aidan P Thompson, Mitchell A Wood, et al. Performance and cost assessment of machine learning interatomic potentials. *The Journal of Physical Chemistry A*, 124(4):731–745, 2020.
- [7] Tim Mueller, Alberto Hernandez, and Chuhong Wang. Machine learning for interatomic potential models. *The Journal of Chemical Physics*, 152(5):050902, 02 2020. ISSN 0021-9606. doi:10.1063/1.5126336. URL <https://doi.org/10.1063/1.5126336>.
- [8] N. Artrith et al. Best practices in machine learning for chemistry. *Nat. Chem.*, 13:505–508, 2021.
- [9] Anh Tran, Julien Tranchida, Tim Wildey, and Aidan P Thompson. Multi-fidelity machine-learning with uncertainty quantification and bayesian optimization for materials design: Application to ternary random alloys. *The Journal of Chemical Physics*, 153(7), 2020.

-
- [10] Yuge Hu, Joseph Musielewicz, Zachary W. Ulissi, and Andrew J. Medford. Robust and scalable uncertainty estimation with conformal prediction for machine-learned interatomic potentials. *Machine Learning: Science and Technology*, 3, 12 2022. ISSN 26322153. doi:10.1088/2632-2153/aca7b1.
- [11] Jonas Busk, Mikkel N. Schmidt, Ole Winther, Tejs Vegge, and Peter Bjørn Jørgensen. Graph neural network interatomic potential ensembles with calibrated aleatoric and epistemic uncertainty on energy and forces. *Phys. Chem. Chem. Phys.*, 25:25828–25837, 2023. doi:10.1039/D3CP02143B. URL <http://dx.doi.org/10.1039/D3CP02143B>.
- [12] Emil Annevelink and Venkatasubramanian Viswanathan. Statistical methods for resolving poor uncertainty quantification in machine learning interatomic potentials. *arXiv preprint arXiv:2308.15653*, 2023.
- [13] Maksim Kulichenko, Kipton Barros, Nicholas Lubbers, Ying Wai Li, Richard Messerly, Sergei Tretiak, Justin S Smith, and Benjamin Nebgen. Uncertainty-driven dynamics for active learning of interatomic potentials. *Nature computational science*, 3(3):230–239, 2023.
- [14] Mads-Peter Verner Christiansen, Nikolaj Rønne, and Bjørk Hammer. Efficient ensemble uncertainty estimation in gaussian processes regression. *Machine Learning: Science and Technology*, 5(4):045029, 2024.
- [15] Matthias Kellner and Michele Ceriotti. Uncertainty quantification by direct propagation of shallow ensembles. *Machine Learning: Science and Technology*, 5(3):035006, 2024.
- [16] Mingjian Wen and Ellad B Tadmor. Uncertainty quantification in molecular simulations with dropout neural network potentials. *npj computational materials*, 6(1):124, 2020.
- [17] Yuge Hu, Joseph Musielewicz, Zachary W Ulissi, and Andrew J Medford. Robust and scalable uncertainty estimation with conformal prediction for machine-learned interatomic potentials. *Machine Learning: Science and Technology*, 3(4):045028, 2022.
- [18] Albert Zhu, Simon Batzner, Albert Musaelian, and Boris Kozinsky. Fast uncertainty estimates in deep learning interatomic potentials. *The Journal of Chemical Physics*, 158(16), 2023.
- [19] Jenna A Billbre, Jesun S Firoz, Mal-Soon Lee, and Sutanay Choudhury. Uncertainty quantification for neural network potential foundation models. *npj Computational Materials*, 11(1):109, 2025.
- [20] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.
- [21] Gabriele Scalia, Colin A. Grambow, Barbara Pernici, Yi Pei Li, and William H. Green. Evaluating scalable uncertainty estimation methods for deep learning-based molecular property prediction. *Journal of Chemical Information and Modeling*, 60:2697–2717, 6 2020. ISSN 1549960X. doi:10.1021/acs.jcim.9b00975.
- [22] Albert Zhu, Simon Batzner, Albert Musaelian, and Boris Kozinsky. Fast uncertainty estimates in deep learning interatomic potentials. *Journal of Chemical Physics*, 158, 4 2023. ISSN 10897690. doi:10.1063/5.0136574.
- [23] Aik Rui Tan, Shingo Urata, Samuel Goldman, Johannes CB Dietschreit, and Rafael Gómez-Bombarelli. Single-model uncertainty quantification in neural network potentials does not consistently outperform model ensembles. *npj Computational Materials*, 9(1):225, 2023.
- [24] Zeynep Sumer, James L. McDonagh, Clyde Fare, Ravikanth Tadikonda, Viktor Zolyomi, David Bray, and Edward Pyzer-Knapp. Providing machine learning potentials with high quality uncertainty estimates, 2025. URL <https://arxiv.org/abs/2501.05250>.
- [25] Leonid Kahle and Federico Zipoli. Quality of uncertainty estimates from neural network potential ensembles. *Physical Review E*, 105, 1 2022. ISSN 24700053. doi:10.1103/PhysRevE.105.015311.
- [26] Jesús Carrete, Hadrián Montes-Campos, Ralf Wanzenböck, Esther Heid, and Georg K.H. Madsen. Deep ensembles vs committees for uncertainty estimation in neural-network force fields: Comparison and application to active learning. *Journal of Chemical Physics*, 158, 5 2023. ISSN 10897690. doi:10.1063/5.0146905.
- [27] Shuaihua Lu, Luca M. Ghiringhelli, Christian Carbogno, Jinlan Wang, and Matthias Scheffler. On the uncertainty estimates of equivariant-neural-network-ensembles interatomic potentials, 2023. URL <https://arxiv.org/abs/2309.00195>.
- [28] David J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3): 448–472, 05 1992. ISSN 0899-7667. doi:10.1162/neco.1992.4.3.448. URL <https://doi.org/10.1162/neco.1992.4.3.448>.
- [29] Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun. Hands-on bayesian neural networks—a tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, 17(2): 29–48, 2022. doi:10.1109/MCI.2022.3155327.

-
- [30] Stephen Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng, editors. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 1st edition, 2011. doi:10.1201/b10905. URL <https://doi.org/10.1201/b10905>.
- [31] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.
- [32] Nongnuch Artrith and Alexander Urban. An implementation of artificial neural-network potentials for atomistic materials simulations: Performance for tio2. *Computational Materials Science*, 114:135–150, 3 2016. ISSN 09270256. doi:10.1016/j.commatsci.2015.11.047.
- [33] Jon López-Zorrilla, Xabier M. Aretxabaleta, In Won Yeu, Iñigo Etxebarría, Hegoi Manzano, and Nongnuch Artrith. `ænet-pytorch`: A gpu-supported implementation for machine learning atomic potentials training. *Journal of Chemical Physics*, 158, 4 2023. ISSN 10897690. doi:10.1063/5.0146803.
- [34] Luis Basora, Arthur Viens, Manuel Arias Chao, and Xavier Olive. A benchmark on uncertainty quantification for deep learning prognostics. *Reliability Engineering & System Safety*, 253:110513, 2025. ISSN 0951-8320. doi:<https://doi.org/10.1016/j.res.2024.110513>. URL <https://www.sciencedirect.com/science/article/pii/S0951832024005854>.
- [35] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep Universal Probabilistic Programming. *Journal of Machine Learning Research*, 2018.
- [36] Hippolyt Ritter and Theofanis Karaletsos. Tyxe: Pyro-based bayesian neural nets for pytorch. *Proceedings of Machine Learning and Systems*, 4:398–413, 2022.
- [37] Gabriel Loaiza-Ganem, Valentin Vilecroze, and Yixin Wang. Deep ensembles secretly perform empirical bayes, 2025. URL <https://arxiv.org/abs/2501.17917>.
- [38] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, April 2017. ISSN 1537-274X. doi:10.1080/01621459.2017.1285773. URL <http://dx.doi.org/10.1080/01621459.2017.1285773>.
- [39] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.
- [40] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *the Journal of machine Learning research*, 14(1):1303–1347, 2013.
- [41] Wray L Buntine. Operations for learning with graphical models. *Journal of artificial intelligence research*, 2: 159–225, 1994.
- [42] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL <https://arxiv.org/abs/1312.6114>.
- [43] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems*, 28, 2015.
- [44] Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. Flipout: Efficient pseudo-independent weight perturbations on mini-batches, 2018. URL <https://arxiv.org/abs/1803.04386>.
- [45] Sebastian Farquhar, Michael A Osborne, and Yarin Gal. Radial bayesian neural networks: Beyond discrete support in large-scale bayesian deep learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1352–1362. PMLR, 2020.
- [46] William Falcon and The PyTorch Lightning team. PyTorch Lightning, March 2019. URL <https://github.com/Lightning-AI/lightning>.
- [47] Kevin Tran, Willie Neiswanger, Junwoong Yoon, Qingyang Zhang, Eric Xing, and Zachary W. Ulissi. Methods for comparing uncertainty quantifications for material property predictions. *Machine Learning: Science and Technology*, 1(2), 2020. ISSN 26322153. doi:10.1088/2632-2153/ab7e1a.
- [48] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2796–2804. PMLR, 10–15 Jul 2018. URL <https://proceedings.mlr.press/v80/kuleshov18a.html>.
- [49] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.