ACTIVE AND TRANSFER LEARNING WITH PARTIALLY BAYESIAN NEURAL NETWORKS FOR MATERIALS AND CHEMICALS

Anonymous authors
Paper under double-blind review

ABSTRACT

Active learning, an iterative process of selecting the most informative data points for exploration, is crucial for efficient characterization of materials and chemicals property space. Neural networks excel at predicting these properties but lack the uncertainty quantification needed for active learning-driven exploration. Fully Bayesian neural networks, in which weights are treated as probability distributions inferred via advanced Markov Chain Monte Carlo methods, offer robust uncertainty quantification but at high computational cost. Here, we show that partially Bayesian neural networks (PBNNs), where only selected layers have probabilistic weights while others remain deterministic, can achieve accuracy and uncertainty estimates on active learning tasks comparable to fully Bayesian networks at lower computational cost. Furthermore, by initializing prior distributions with weights pre-trained on theoretical calculations, we demonstrate that PBNNs can effectively leverage computational predictions to accelerate active learning of experimental data. We validate these approaches on both molecular property prediction and materials science tasks, establishing PBNNs as a practical tool for active learning with limited, complex datasets.

1 INTRODUCTION

Active learning (AL) (Cohn et al., 1996; Settles, 2009) optimizes exploration of large parameter spaces by strategically selecting which experiments or simulations to conduct, reducing resource consumption and potentially accelerating scientific discovery (Cao et al., 2024; Lookman et al., 2019; Wang et al., 2022; Xu et al., 2023; Slautin et al., 2024; Ziatdinov et al., 2022). A key component of this approach is a surrogate machine learning (ML) model, which approximates an unknown functional relationship between structure or process parameters and target properties. At each step, the model uses the information gathered from previous measurements to update its "understanding" of these relationships and identify the next combinations of parameters likely to yield valuable information. The success of this approach critically depends on reliable uncertainty quantification (UQ) in the underlying ML models.

The development of effective ML models for active learning builds upon broader advances in machine learning across materials and chemical sciences, tackling problems including phase stability (Arróyave, 2022; Peivaste et al., 2023; Liu et al., 2024), thermal conductivity (Huang et al., 2023; Luo et al., 2023; Barua et al., 2024; Carrete et al., 2014), glass transition temperatures (Liu & Su, 2024; Zhang et al., 2023; Armeli et al., 2023; Galeazzo & Shiraiwa, 2022; Uddin & Fan, 2024), dielectric properties (Hu et al., 2024; Dong et al., 2021; Grumet et al., 2024; Shimano et al., 2023), and more (Morgan & Jacobs, 2020; Chong et al., 2023; Zhong et al., 2022; Schmidt et al., 2019). However, traditional ML models often lack robust UQ, posing challenges for their application in AL workflows. Moreover, many of them are trained on computational data, such as density functional theory calculations, and generalization to experimental workflows in physical labs, where data are often sparse, noisy, and costly to acquire, is often non-trivial and requires predictions with reliable coverage probabilities.

Gaussian Process (GP) (Rasmussen & Williams, 2005; Snoek et al., 2012; Gramacy, 2020) is an ML approach that provides mathematically-grounded UQ and has become a popular choice for

scientific applications, including AL frameworks (Deringer et al., 2021; Ziatdinov et al., 2022). However, GPs struggle with high-dimensional data, discontinuities, and non-stationarities, which are common in physical science problems. Deep kernel learning (DKL) (Calandra et al., 2016; Wilson et al., 2016a;b) attempts addressing these issues by combining neural network representation learning with GP-based UQ. While DKL has shown promise in chemistry and materials science (Singh & Hernandez-Lobato, 2024; Duhrkop, 2022; Valleti et al., 2024), it is still limited by GP scalability in feature space, potential mode collapse, and conflicting optimization dynamics between its GP and neural network components (Ober et al., 2021). These limitations highlight the need for further advancement of methods to support AL in non-trivial materials design and discovery tasks.

Bayesian neural networks (BNNs), where all network weights are treated as probability distributions rather than scalar values (Titterington, 2004; Lampinen & Vehtari, 2001), offer a promising approach that combines powerful representation learning capabilities with reliable UQ. By maintaining a distribution over network parameters rather than point estimates, BNNs naturally account for model uncertainty, and are particularly effective for smaller and noisier datasets. However, reliable Bayesian inference requires computationally intensive sampling methods, making fully Bayesian neural networks prohibitively expensive for many practical applications. In this work, we explore *partially* Bayesian neural networks (PBNNs) for active learning of molecular and materials properties. We show that by making strategic choices about which layers are treated probabilistically we can achieve performance on active learning tasks comparable to fully Bayesian neural networks at significantly reduced computational cost. Furthermore, we demonstrate how PBNNs can be enhanced through transfer learning by initializing their prior distributions from weights pre-trained on computational data. We validate these approaches on both molecular property prediction and materials science tasks, establishing PBNNs as a practical tool for active learning with limited, complex datasets.

2 METHODS

We have examined the potential of BNNs and PBNNs for active and transfer learning on several benchmark datasets. Descriptions of the datasets, as well as details regarding our active learning workflow, are given in Appendices A.1 and A.2, respectively.

2.1 BAYESIAN NEURAL NETWORKS

In conventional, non-Bayesian NNs, network weights θ are optimized to minimize a specified loss function, resulting in a deterministic, single-point prediction for each new input. Due to their architectural flexibility they can be powerful function approximators, but are known to suffer from overfitting on small or noisy datasets and overconfidence on out-of-distribution inputs (Nguyen et al., 2015; Hendrycks & Gimpel, 2017; Lakshminarayanan et al., 2017). In contrast, in BNNs the weights θ are treated as random variables with a prior distribution $p(\theta)$. This not only helps reduce overfitting, but also provides robust prediction uncertainties. Given a dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$, a BNN is defined by its probabilistic model:

$$\textit{Weights}\text{: } \theta \sim p(\theta) \quad \text{(typically } \mathcal{N}(0,1)) \tag{1}$$

$$\textit{Noise}\text{: } \sigma \sim p(\sigma) \quad \text{(typically Half-Normal}(0,1)) \tag{2}$$

$$\textit{Likelihood}\text{: } y_i|x_i, \theta, \sigma \sim \mathcal{N}(g(x_i;\theta), \sigma^2) \tag{3}$$

where $g(x_i;\theta)$ represents the neural network function mapping inputs to outputs using weights θ. While we focus on normal likelihoods here for regression tasks, the framework naturally extends to other distributions (e.g., Bernoulli for classification, Poisson for count data) depending on the problem domain. The posterior predictive distribution for new input x^* is then given by

$$p(y|x^*, \mathcal{D}) = \int_{\theta,\sigma} p(y|x^*, \theta, \sigma) p(\theta, \sigma|\mathcal{D}) d\theta d\sigma \tag{4}$$

This predictive distribution can be interpreted as an infinite ensemble of networks, with each network's contribution to the overall prediction weighted by the posterior probability of its weights given the training data. Unfortunately, the posterior $p(\theta, \sigma|\mathcal{D})$ in Eq. (4) is typically intractable. It

is therefore common to use Markov Chain Monte Carlo (MCMC) (Hastings, 1970) or variational inference (Blei et al., 2017) techniques to approximate the posterior. The advanced MCMC methods, such as Hamiltonian Monte Carlo (HMC) (Betancourt, 2018), generally provide higher accuracy than variational methods for complex posterior distributions. Here, we employ the No-U-Turn Sampler (NUTS) extension of the HMC, which constructs a Markov chain of network weight and noise samples that converges to the posterior distribution $p(\theta, \sigma|\mathcal{D})$. The algorithm uses Hamiltonian dynamics with the network weights as position variables, employing leapfrog integration and adaptively determining the trajectory length to efficiently explore the parameter space (Homan & Gelman, 2014). The predictive mean (μ_{post}) and predictive variance (U_{post}) at new data points are then given by:

$$\mu^{post} = \frac{1}{N}\sum_{i=1}^{N} g(x^*; \theta_i) \tag{5}$$

$$U^{post} = \frac{1}{N}\sum_{i=1}^{N} (y_i^* - \mu^{post})^2 \tag{6}$$

$$y_i^* \sim \mathcal{N}(g(x^*; \theta_i), \sigma_i^2) \tag{7}$$

where y_i^* is a single sample from the model posterior at new input x^*, $\{\theta_i, \sigma_i\}_{i=1}^N$ are samples from the MCMC chain approximating $p(\theta, \sigma|\mathcal{D})$, and N is the total number of MCMC samples. Note that U^{post} naturally combines both epistemic uncertainty (from the variation in network predictions across different weight samples θ_i) and aleatoric uncertainty (from the noise terms σ_i), providing a comprehensive measure of predictive uncertainty.

2.1.1 Partially Bayesian neural networks

Even with sampling methods, full BNNs can be computationally expensive for reasonably-sized datasets, in terms of number of samples or feature dimensions. Variational inference, a common approximation method for BNNs, aims to alleviate these costs but often struggles with limited expressivity, underestimation of uncertainty, and sensitivity to initialization and hyperparameters, which degrades its performance on real-world tasks. To leverage the representational power and computational efficiency of deterministic NNs *and* the advantages of BNNs, we explore partially Bayesian neural networks (PBNNs), where only a selected number of layers are probabilistic and all other layers are deterministic. Building upon existing research that proposed usage of selectively stochastic layers (Sharma et al., 2023; Harrison et al., 2024), our work specifically investigates the potential of PBNNs in active and transfer learning contexts, with a focus on molecular and materials science datasets.

The PBNNs are trained in two stages. First, it trains a deterministic neural network, incorporating stochastic weight averaging (SWA) (Izmailov et al., 2019) at the end of the training trajectory to enhance robustness against noisy training objectives. Second, the probabilistic component is introduced by selecting a subset of layers and using the corresponding pre-trained weights to initialize prior distributions for this subset, while keeping all remaining weights frozen. HMC/NUTS sampling is then applied to derive posterior distributions for the selected subset. Finally, predictions are made by combining both the probabilistic and deterministic components. See Algorithm 1 and Figure 1 for more details. In certain scenarios, such as autonomous experiments, the entire training process needs to be performed in an end-to-end manner. In these cases, it is crucial to avoid overfitting in the deterministic component, as there will be no human oversight to evaluate its results before transitioning to the probabilistic part. To address this, we incorporate a MAP prior, modeled as a Gaussian penalty, into the loss function during deterministic training. All the PBNNs were implemented via the NeuroBayes package[1].

In this work, we have investigated PBNNs of multilayer perceptron (MLP) architecture consisting of five layers: four utilize non-linear activation functions, such as the sigmoid linear unit, while the final (output) layer contains a single neuron without a non-linear activation, as is typical for regression tasks. As there are multiple ways to select probabilistic layers for the PBNNs, we have evaluated the effects of setting different combinations of probabilistic layers as shown in Figure 2.

[1] https://github.com/ziatdinovmax/NeuroBayes

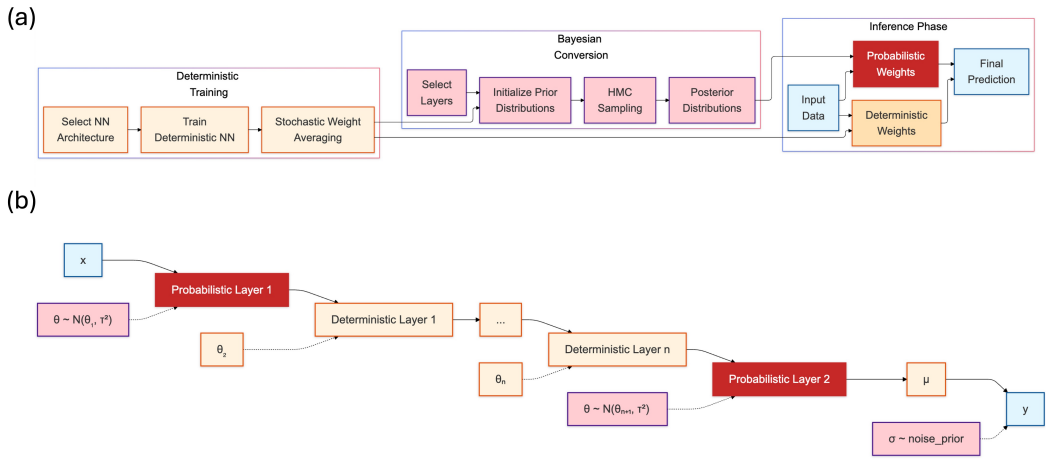

Figure 1: (a) Schematic illustration of Partially Bayesian Neural Network (PBNN) operation. First, we train a deterministic neural network, incorporating stochastic weight averaging to enhance robustness against noisy training objectives. Second, the probabilistic component is introduced by selecting a subset of layers and using the corresponding pre-trained weights to initialize prior distributions for this subset, while keeping all remaining weights frozen. HMC/NUTS sampling is then applied to derive posterior distributions for the selected subset. Finally, predictions are made by combining both the probabilistic and deterministic components. (b) Schematic illustration of flow through a PBNN model alternating probabilistic and deterministic processing stages.

Algorithm 1 Partially Bayesian Neural Network Training

Require:
Input data $X \in \mathbb{R}^{n \times d}$, targets $y \in \mathbb{R}^n$
Deterministic neural network architecture g_θ
Set of probabilistic layers $\mathcal{L}$
Optional: Custom SWA collection protocol ψ
Optional: Custom prior width τ for probabilistic weights
† Deterministic training hyperparameters follow typical deep learning practices
‡ Probabilistic training parameters follow standard Bayesian inference practices

Initialize network parameters θ
Initialize empty weights collection $\mathcal{W} = \{\}$
for epoch $e = 1$ to E **do**
    η_e, collect $= \psi(e, E)$
    Update θ using SGD: $\theta \leftarrow \theta - \eta_e \nabla \mathcal{L}(\theta)$
    if collect **then**
        Add current weights to collection: $\mathcal{W} = \mathcal{W} \cup \{\theta\}$
    end if
end for
Compute averaged weights $\theta_{det} = \frac{1}{|\mathcal{W}|} \sum_{\theta \in \mathcal{W}} \theta$
// Run HMC/NUTS sampler for posterior inference
for each layer l in network **do**
    if l is probabilistic **then**
        Set prior $p(\theta_l) = \mathcal{N}(\theta_{det,l}, \tau)$
        Sample weights $\theta_l \sim p(\theta_l)$
    else
        Set weights $\theta_l = \theta_{det,l}$
    end if
end for
Calculate network output $\mu = g_\theta(X)$
Sample observation noise $\sigma \sim p(\sigma)$
Score observations $y \sim \mathcal{N}(\mu, \sigma^2)$
return Posterior samples of probabilistic weights and noise parameter

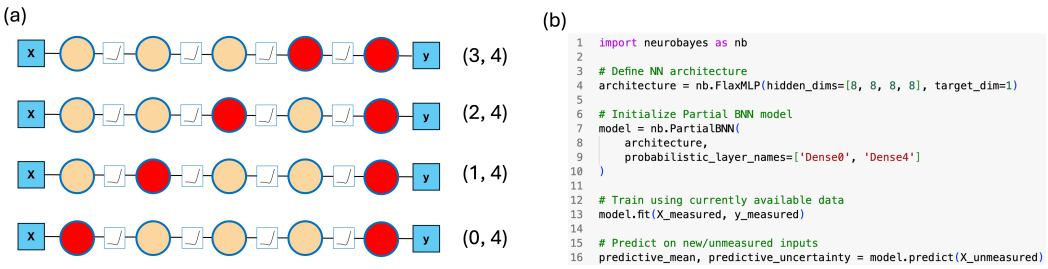

Figure 2: (a) Schematic representation of the partially Bayesian MLP employed in this study. The model consists of five layers: four utilize non-linear activation functions, such as the sigmoid linear unit, while the final (output) layer contains a single neuron without a non-linear activation, as is typical for regression tasks. Circles filled with red denote stochastic layers, while orange filled circles represent deterministic layers. Note that the single output neuron is always made probabilistic, as it often improves training stability. (b) Code snippet illustrating a single train-predict step with PBNN (0, 4).

3 Results and Discussion

3.1 Active learning on molecular datasets

We first investigate the effectiveness of different PBNNs for AL on the standard molecular benchmark datasets. Figures 3(a) and 3(b) show RMSE, NLPD, and coverage probability as a function of AL exploration step for ESOL and FreeSolv, respectively. We see that the accuracy and quality of the uncertainties improve with AL for all PBNNs, as demonstrated by *i*) decreasing RMSE and NLPD and *ii*) increasing coverage over time for all models. Across all metrics for both datasets, making earlier layers probabilistic proves more effective, with PBNN(0,4) approaching the accuracy of a Full BNN. Furthermore, PBNN(0,4) exhibits a relatively stable decrease in NLPD and increase in coverage throughout the AL process, similar to Full BNN. In contrast, configurations where the probabilistic layer is moved away from the first hidden layer, PBNN(1,4), (2,4), and (3,4), show strong oscillatory behavior in NLPD and coverage metrics, suggesting that uncertainty propagation becomes unstable when probabilistic layers are placed in later hidden layers. This shows that, at least within the standard MLP architecture employed here, capturing uncertainty in the first feature transformation layer, combined with a probabilistic output layer, is more effective, both in terms of performance and reliability. In addition, it decreased the overall computational time by nearly a factor of four. Notably, with only a fraction of points explored, AL with PBNN achieves accuracy either comparable to (ESOL) or better than (FreeSolv) that obtained using standard 80:20 or 90:10 train-test splits with standard deterministic ML models (Wu et al., 2018).

3.2 Active learning on materials datasets

Next, we follow a similar analysis for the two materials datasets, Steel fatigue (NIMS) and Conductivity (HTEM), as shown in Figure 4. We observe overall similar trends to the molecular datasets (decreasing RMSE and NLPD and increasing coverage), although we see a much stronger difference between the different PBNNs in the uncertainty metrics, with smaller difference in RMSE across different selections of probabilistic layers. We also do not observe the clean monotonic trends that we observed with the molecular datasets for NLPD and Coverage on the Steel fatigue (NIMS) dataset. This could be due to a variety of factors, but we suspect that this is largely due to differences in the types of input features. While the molecular datasets utilized SMILES-derived descriptors as their input features, the materials datasets contained experimental parameters as their input features, which may not be as predictive of the target properties as the structural SMILES-based descriptors. There could also be a difference in experimental noise between the molecular and materials datasets, as it is well known that values of the materials target properties, fatigue strength and electrical conductivity, are sensitive to experimental variations in their measurement, whereas measurements of hydration free energy and aqueous solubility are relatively standardized.

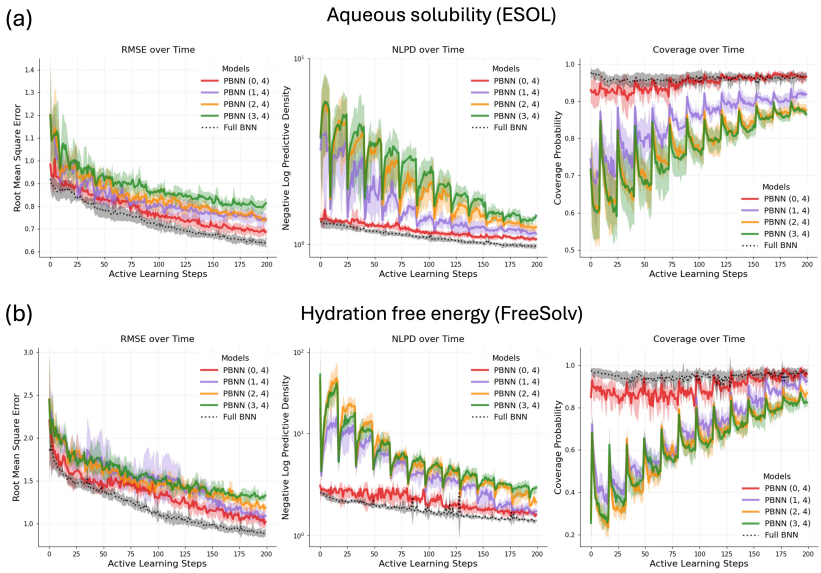

Figure 3: Comparison of Partially Bayesian Neural Networks (PBNNs) and fully Bayesian neural network (Full BNN) on molecular property prediction tasks. (a) Aqueous solubility prediction (ESOL database) and (b) hydration free energy prediction (FreeSolv database). Each PBNN configuration PBNN (i, 4) has two probabilistic layers: one at position i (counting from 0) and one at the output. Shaded areas represent a standard deviation across five different random seeds.

Despite these domain-specific variations, the results across both molecular and materials domains support the emerging general principle that making the first hidden and the output layers probabilistic is more effective than doing so for intermediate or final layers. We would also like to emphasize that we used the same MLP architecture and training parameters (SGD learning rate and iterations for the deterministic component, warmup steps and samples for NUTS in the probabilistic component) across all four datasets. This demonstrates that PBNNs can be relatively robust to hyperparameter selection, a valuable characteristic for practical applications as it minimizes the need for extensive dataset-specific tuning.

3.3 Convergence Diagnostics

We next discuss convergence diagnostics for PBNN models during active learning. A popular choice for convergence diagnostics in Bayesian inference is the Gelman-Rubin statistic ('R-hat'), which provides a measure of convergence for each model parameter (Gelman & Rubin, 1992). However, for Bayesian neural networks, where the parameter space is high-dimensional, examining individual parameter convergence becomes impractical. Instead, we analyzed the distribution of R-hat values across all parameters and found that for the majority of weights (95–99%, depending on dataset), these values lie within acceptable ranges between 1.0 and 1.1 (Brooks & Gelman, 1998). While layer-wise or module-wise convergence analysis is also possible for complex architectures, we opted for global parameter statistics due to the relatively simple network structure in this study. See Appendix A.3 for more details.

We note that in active learning-based autonomous science tasks, reliable convergence diagnostics play an important role in ensuring the autonomous system performance. The R-hat statistic can therefore serve as an automated quality check, triggering specific actions when convergence issues are detected: for example, if a high proportion ($> 10\%$) of parameters display R-hat values outside the acceptable range, the system can employ various convergence improvement heuristics. These include increasing the number of warm-up states, trying different parameter initialization schemes, or adjusting prior distributions. If issues persist after these interventions, the system can flag the experiment for human review, ensuring reliability of the autonomous decision-making process.

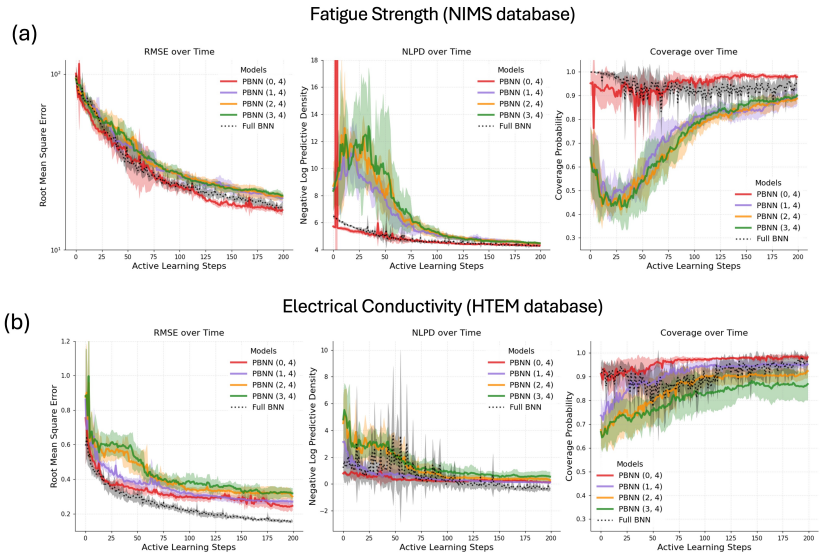

Figure 4: Comparison of Partially Bayesian Neural Networks (PBNNs) and fully Bayesian neural network (Full BNN) on materials property prediction tasks. (a) Fatigue strength prediction (NIMS database) and (b) electrical conductivity prediction (HTEM database). Each PBNN configuration PBNN (i, 4) has two probabilistic layers: one at position i (counting from 0) and one at the output. Shaded areas represent a standard deviation across five different random seeds.

3.4 TRANSFER LEARNING

Transfer learning (TL) is particularly valuable when data is limited and difficult to acquire, as is often the case in experimental materials science and chemistry. For deterministic NNs, TL is performed by initializing the network parameters with those of a pre-trained network. Most often the target NN's parameters are still optimized for the task at-hand via backpropagation, which is referred to as fine-tuning. In the context of BNNs, TL can be done through a selection of prior distributions over the weights, where the priors incorporate some domain knowledge. Here, we use the weights of a deterministic model trained in a *computational* space to initialize the prior distributions by setting their means to the corresponding pre-trained weights, thereby transferring domain knowledge to a (P)BNN operating in the *experimental* space. We can do this for the entire model or only for some parts (layers). We can also specify a "degree of trust" in the theory by selecting appropriate standard deviations for these distributions: wider distributions indicate less confidence in the computational model, while narrower ones encode stronger confidence. Here, we examine how this simulation-to-experiment transfer learning affects AL with (P)BNNs. The process involves first training a deterministic NN on simulation data, then using its weights to inform the (P)BNN surrogate model that guides active learning on experimental data.

We start with Noisy-FreeSolv dataset. Here the deterministic neural network is pre-trained on computational data from molecular dynamics simulations, whereas experimental data is augmented with synthetic noise to create a more challenging test case for our models. For this study, we made the last two hidden layers and the output layer probabilistic, with priors initialized at values of weights from the corresponding pre-trained deterministic neural network. Figure 5 shows the performance of PBNN with theory-informed priors for different prior widths (τ). While all prior widths demonstrate good performance, intermediate width ($\tau = 0.5$) achieves slightly better RMSE and NLPD values compared to tighter ($\tau = 0.1$) or wider ($\tau = 1.0$) priors, suggesting an optimal balance between leveraging theoretical knowledge and adapting to experimental data. Comparing pre-trained and standard priors at $\tau = 1.0$, we observe that theory-informed priors lead to substantially better performance across all metrics. The improvement is particularly pronounced in NLPD and coverage, where standard priors show high uncertainty and unstable behavior throughout the active learning process.

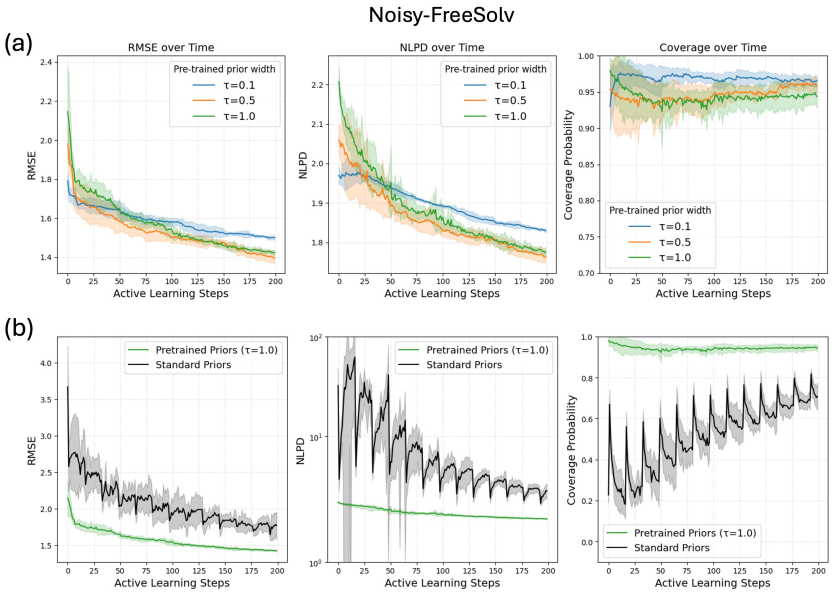

Figure 5: Transfer learning with pre-trained PBNNs applied to noisy FreeSolv dataset. (a) RMSE, NLPD, and coverage probability for different prior widths (τ). (b) Comparing the performance of pre-trained priors ($\tau = 1.0$) against standard priors. Shaded areas represent a standard deviation across five different random seeds.

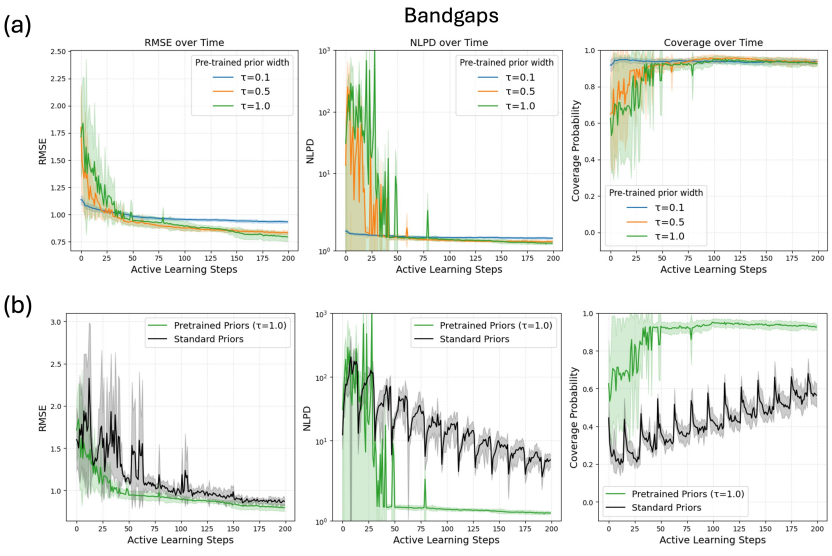

Figure 6: Transfer learning with pre-trained PBNNs applied to Bandgaps dataset. (a) RMSE, NLPD, and coverage probability for different prior widths (τ). (b) Comparing the performance of pre-trained priors ($\tau = 1.0$) against standard priors. Shaded areas represent a standard deviation across five different random seeds.

Finally, we analyze bandgaps of non-metals, where priors are pre-trained on density functional theory (DFT) calculations. The results shown in Figure 6 demonstrate that among different prior widths, there is a clear trade-off: the tight prior ($\tau = 0.1$) shows stable but limited improvement, suggesting it constrains the model too closely to DFT predictions, while wider priors ($\tau = 0.5$ and $\tau = 1.0$) show initial oscillations but ultimately achieve better RMSE through greater adaptation to experimental data. This suggests that one can in principle apply dynamic adjustment: impose a strong belief in the theoretical model initially, and then, as more data becomes available, gradually relax it, allowing the data to speak for itself. Comparing pre-trained and standard priors at $\tau = 1.0$, we observe similar trends to the FreeSolv dataset. The advantage of pre-trained priors is particularly pronounced in the early stages of active learning, where in the first 50 steps they achieve significantly lower RMSE and better calibrated uncertainties compared to standard priors, indicating more efficient use of limited experimental data. While both approaches eventually converge to similar RMSE values, the benefits of pre-trained priors persist in uncertainty quantification throughout the entire process, maintaining substantially better coverage probability.

4 Conclusion

In this work, we explored the capabilities of partially Bayesian neural networks (PBNNs) in active learning tasks. Within the MLP architectures deployed here, we found that the choice of which layers are made probabilistic significantly impacts performance, with early layers providing better and more stable uncertainty estimates - a finding that held consistently across studied molecular and materials datasets. Notably, PBNNs with probabilistic first layer achieved performance comparable to fully Bayesian networks while requiring substantially fewer computational resources. We further enhanced PBNN performance through transfer learning by initializing priors using theoretical models, which proved particularly beneficial in the early stages of active learning. Our analysis revealed an important trade-off in prior width selection: tight priors ensure stability but may constrain the model too closely to theoretical predictions, while wider priors enable better adaptation to experimental data. Across both studied systems, theory-informed priors led to better calibrated uncertainties and more efficient data utilization. Overall, this work demonstrates the feasibility of PBNNs for materials science and chemistry, particularly in the context of AL for limited, complex datasets.

5 Code and Data Availability

The repository containing code and data supporting the paper's findings, together with additional implementation details, will be specified upon acceptance.

Acknowledgments

This work was supported by the Laboratory Directed Research and Development Program at Pacific Northwest National Laboratory, a multiprogram national laboratory operated by Battelle for the U.S. Department of Energy.

References

A. Agrawal, Deshpande P. D., G. P. Basavarsu, A. N. Choudhary, and S. R. Kalidindi. Exploration of data science techniques to predict fatigue strength of steel from composition and processing parameters. *Integr Mater Manuf Innov*, 3:90–108, 2014. doi: 10.1186/2193-9772-3-8.

G. Armeli, J. H. Peters, and T. Koop. Machine-learning-based prediction of the glass transition temperature of organic compounds using experimental data. *ACS Omega*, 8(13):12298–12309, 2023. ISSN 2470-1343 (Electronic) 2470-1343 (Linking). doi: 10.1021/acsomega.2c08146. URL https://www.ncbi.nlm.nih.gov/pubmed/37033862.

Raymundo Arróyave. Phase stability through machine learning. *Journal of Phase Equilibria and Diffusion*, 43(6):606–628, 2022. ISSN 1547-7037 1863-7345. doi: 10.1007/s11669-022-01009-9.

Nikhil K. Barua, Evan Hall, Yifei Cheng, Anton O. Oliynyk, and Holger Kleinke. Interpretable machine learning model on thermal conductivity using publicly available datasets and our internal lab dataset. *Chemistry of Materials*, 36(14):7089–7100, 2024. ISSN 0897-4756 1520-5002. doi: 10.1021/acs.chemmater.4c01696.

Michael Betancourt. A conceptual introduction to hamiltonian monte carlo, 2018. URL https://arxiv.org/abs/1701.02434.

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.

Stephen P. Brooks and Andrew Gelman. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455, 1998. doi: 10.1080/10618600.1998.10474787.

Roberto Calandra, Jan Peters, Carl Edward Rasmussen, and Marc Peter Deisenroth. Manifold gaussian processes for regression. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 3338–3345, 2016. doi: 10.1109/IJCNN.2016.7727626.

Bin Cao, Tianhao Su, Shuting Yu, Tianyuan Li, Taolue Zhang, Jincang Zhang, Ziqiang Dong, and Tong-Yi Zhang. Active learning accelerates the discovery of high strength and high ductility lead-free solder alloys. *Materials & Design*, 241, 2024. ISSN 02641275. doi: 10.1016/j.matdes.2024.112921.

Jesús Carrete, Wu Li, Natalio Mingo, Shidong Wang, and Stefano Curtarolo. Finding unprecedentedly low-thermal-conductivity half-heusler semiconductors via high-throughput materials modeling. *Physical Review X*, 4(1), 2014. ISSN 2160-3308. doi: 10.1103/PhysRevX.4.011019.

Sue Sin Chong, Yi Sheng Ng, Hui-Qiong Wang, and Jin-Cheng Zheng. Advances of machine learning in materials science: Ideas and techniques. *Frontiers of Physics*, 19(1), 2023. ISSN 2095-0462 2095-0470. doi: 10.1007/s11467-023-1325-z.

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *J. Artif. Int. Res.*, 4(1):129–145, 1996. ISSN 1076-9757.

RDKit Contributors. Rdkit: Open-source cheminformatics: https://www.rdkit.org.

John S. Delaney. Esol: Estimating aqueous solubility directly from molecular structure. *Journal of Chemical Information and Computer Sciences*, 44(3):1000–1005, 2004. doi: 10.1021/ci034243x.

Volker L Deringer, Albert P Bartók, Noam Bernstein, David M Wilkins, Michele Ceriotti, and Gábor Csányi. Gaussian process regression for materials and molecules. *Chemical Reviews*, 121(16):10073–10141, 2021.

S. S. Dong, M. Govoni, and G. Galli. Machine learning dielectric screening for the simulation of excited state properties of molecules and materials. *Chem Sci*, 12(13):4970–4980, 2021. ISSN 2041-6520 (Print) 2041-6539 (Electronic) 2041-6520 (Linking). doi: 10.1039/d1sc00503k. URL https://www.ncbi.nlm.nih.gov/pubmed/34163744.

K. Duhrkop. Deep kernel learning improves molecular fingerprint prediction from tandem mass spectra. *Bioinformatics*, 38(Suppl 1):i342–i349, 2022. ISSN 1367-4811 (Electronic) 1367-4803 (Print) 1367-4803 (Linking). doi: 10.1093/bioinformatics/btac260. URL https://www.ncbi.nlm.nih.gov/pubmed/35758813.

Tommaso Galeazzo and Manabu Shiraiwa. Predicting glass transition temperature and melting point of organic compounds via machine learning and molecular embeddings. *Environmental Science: Atmospheres*, 2(3):362–374, 2022. ISSN 2634-3606. doi: 10.1039/d1ea00090j.

Andrew Gelman and Donald B. Rubin. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4):457 – 472, 1992. doi: 10.1214/ss/1177011136. URL https://doi.org/10.1214/ss/1177011136.

Robert Gramacy. *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*. 03 2020. ISBN 9780367815493. doi: 10.1201/9780367815493.

Manuel Grumet, Clara von Scarpatetti, Tomas Bucko, and David A. Egger. Delta machine learning for predicting dielectric properties and raman spectra. *The Journal of Physical Chemistry C*, 128 (15):6464–6470, 2024. ISSN 1932-7455. doi: 10.1021/acs.jpcc.4c00886.

James Harrison, John Willes, and Jasper Snoek. Variational bayesian last layers, 2024. URL https://arxiv.org/abs/2404.11599.

W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks, 2017.

Matthew D. Homan and Andrew Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, January 2014. ISSN 1532-4435.

Yilin Hu, Maokun Wu, Miaojia Yuan, Yichen Wen, Pengpeng Ren, Sheng Ye, Fayong Liu, Bo Zhou, Hui Fang, Runsheng Wang, Zhigang Ji, and Ru Huang. Accurate prediction of dielectric properties and bandgaps in materials with a machine learning approach. *Applied Physics Letters*, 125(15):152905, 10 2024. ISSN 0003-6951. doi: 10.1063/5.0223890. URL https://doi.org/10.1063/5.0223890.

Xiang Huang, Shengluo Ma, C. Y. Zhao, Hong Wang, and Shenghong Ju. Exploring high thermal conductivity polymers via interpretable machine learning with physical descriptors. *npj Computational Materials*, 9(1), 2023. ISSN 2057-3960. doi: 10.1038/s41524-023-01154-w.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization, 2019. URL https://arxiv.org/abs/1803.05407.

Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, and Kristin A. Persson. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1), 2013. ISSN 2166-532X. doi: 10.1063/1.4812323.

Benjamin Kompa, Jasper Snoek, and Andrew L. Beam. Empirical frequentist coverage of deep learning uncertainty quantification procedures. *Entropy*, 23(12):1608, November 2021. ISSN 1099-4300. doi: 10.3390/e23121608. URL http://dx.doi.org/10.3390/e23121608.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

J. Lampinen and A. Vehtari. Bayesian approach for neural networks—review and case studies. *Neural Networks*, 14(3):257–274, 2001.

Chengcheng Liu and Hang Su. Prediction of glass transition temperature of oxide glasses based on interpretable machine learning and sparse data sets. *Materials Today Communications*, 40:109691, 2024. ISSN 2352-4928. doi: https://doi.org/10.1016/j.mtcomm.2024.109691. URL https://www.sciencedirect.com/science/article/pii/S2352492824016726.

Shusen Liu, Brandon Bocklund, James Diffenderfer, Shreya Chaganti, Bhavya Kailkhura, Scott K. McCall, Brian Gallagher, Aurélien Perron, and Joseph T. McKeown. A comparative study of predicting high entropy alloy phase fractions with traditional machine learning and deep neural networks. *npj Computational Materials*, 10(1), 2024. ISSN 2057-3960. doi: 10.1038/s41524-024-01335-1.

Turab Lookman, Prasanna V. Balachandran, Dezhen Xue, and Ruihao Yuan. Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design. *npj Computational Materials*, 5(1), 2019. ISSN 2057-3960. doi: 10.1038/s41524-019-0153-8.

Yufeng Luo, Mengke Li, Hongmei Yuan, Huijun Liu, and Ying Fang. Predicting lattice thermal conductivity via machine learning: a mini review. *npj Computational Materials*, 9(1), 2023. ISSN 2057-3960. doi: 10.1038/s41524-023-00964-2.

D. L. Mobley and J. P. Guthrie. Freesolv: a database of experimental and calculated hydration free energies, with input files. *J Comput Aided Mol Des*, 28(7):711–20, 2014. ISSN 1573-4951 (Electronic) 0920-654X (Print) 0920-654X (Linking). doi: 10.1007/s10822-014-9747-x. URL https://www.ncbi.nlm.nih.gov/pubmed/24928188.

Dane Morgan and Ryan Jacobs. Opportunities and challenges for machine learning in materials science. *Annual Review of Materials Research*, 50(1):71–103, 2020. ISSN 1531-7331 1545-4118. doi: 10.1146/annurev-matsci-070218-010015.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, 2015.

Sebastian W. Ober, Carl E. Rasmussen, and Mark van der Wilk. The promises and pitfalls of deep kernel learning. In Cassio de Campos and Marloes H. Maathuis (eds.), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 1206–1216. PMLR, Jul 2021. URL https://proceedings.mlr.press/v161/ober21a.html.

I. Peivaste, E. Jossou, and A. A. Tiamiyu. Data-driven analysis and prediction of stable phases for high-entropy alloy design. *Sci Rep*, 13(1):22556, 2023. ISSN 2045-2322 (Electronic) 2045-2322 (Linking). doi: 10.1038/s41598-023-50044-0. URL https://www.ncbi.nlm.nih.gov/pubmed/38110634.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005. ISBN 9780262256834. doi: 10.7551/mitpress/3206.001.0001. URL https://doi.org/10.7551/mitpress/3206.001.0001.

Jonathan Schmidt, Mário R. G. Marques, Silvana Botti, and Miguel A. L. Marques. Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials*, 5(1), 2019. ISSN 2057-3960. doi: 10.1038/s41524-019-0221-0.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. URL http://axon.cs.byu.edu/~martinez/classes/778/Papers/settles.activelearning.pdf.

Mrinank Sharma, Sebastian Farquhar, Eric Nalisnick, and Tom Rainforth. Do bayesian neural networks need to be fully stochastic?, 2023. URL https://arxiv.org/abs/2211.06291.

Y. Shimano, A. Kutana, and R. Asahi. Machine learning and atomistic origin of high dielectric permittivity in oxides. *Sci Rep*, 13(1):22236, 2023. ISSN 2045-2322 (Electronic) 2045-2322 (Linking). doi: 10.1038/s41598-023-49603-2. URL https://www.ncbi.nlm.nih.gov/pubmed/38097712.

S. Singh and J. M. Hernandez-Lobato. Deep kernel learning for reaction outcome prediction and optimization. *Commun Chem*, 7(1):136, 2024. ISSN 2399-3669 (Electronic) 2399-3669 (Linking). doi: 10.1038/s42004-024-01219-x. URL https://www.ncbi.nlm.nih.gov/pubmed/38877182.

B. N. Slautin, Y. Liu, H. Funakubo, R. K. Vasudevan, M. Ziatdinov, and S. V. Kalinin. Bayesian conavigation: Dynamic designing of the material digital twins via active learning. *ACS Nano*, 18(36):24898–24908, 2024. ISSN 1936-086X (Electronic) 1936-0851 (Linking). doi: 10.1021/acsnano.4c05368. URL https://www.ncbi.nlm.nih.gov/pubmed/39183496.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf.

D. M. Titterington. Bayesian methods for neural networks and related models. *Statistical Science*, 19(1), 2004. ISSN 0883-4237. doi: 10.1214/088342304000000099.

M. J. Uddin and J. Fan. Interpretable machine learning framework to predict the glass transition temperature of polymers. *Polymers (Basel)*, 16(8), 2024. ISSN 2073-4360 (Electronic) 2073-4360 (Linking). doi: 10.3390/polym16081049. URL https://www.ncbi.nlm.nih.gov/pubmed/38674969.

Mani Valleti, Rama K. Vasudevan, Maxim A. Ziatdinov, and Sergei V. Kalinin. Deep kernel methods learn better: from cards to process optimization. *Machine Learning: Science and Technology*, 5 (1), 2024. ISSN 2632-2153. doi: 10.1088/2632-2153/ad1a4f.

Alex Wang, Haotong Liang, Austin McDannald, Ichiro Takeuchi, and Aaron Gilad Kusne. Benchmarking active learning strategies for materials optimization and discovery. *Oxford Open Materials Science*, 2(1), 2022. ISSN 2633-6979. doi: 10.1093/oxfmat/itac006.

Logan Ward, Ankit Agrawal, Alok Choudhary, and Christopher Wolverton. A general-purpose machine learning framework for predicting properties of inorganic materials. *npj Computational Materials*, 2(1):1–7, 2016.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning. In Arthur Gretton and Christian C. Robert (eds.), *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 370–378, Cadiz, Spain, 09–11 May 2016a. PMLR. URL https://proceedings.mlr.press/v51/wilson16.html.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Stochastic variational deep kernel learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 2594–2602, Red Hook, NY, USA, 2016b. Curran Associates Inc. ISBN 9781510838819.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. Moleculenet: A benchmark for molecular machine learning, 2018. URL https://arxiv.org/abs/1703.00564.

Pengcheng Xu, Xiaobo Ji, Minjie Li, and Wencong Lu. Small data machine learning in materials science. *npj Computational Materials*, 9(1), 2023. ISSN 2057-3960. doi: 10.1038/s41524-023-01000-z.

A. Zakutayev, N. Wunder, M. Schwarting, J. D. Perkins, R. White, K. Munch, W. Tumas, and C. Phillips. An open experimental database for exploring inorganic materials. *Sci Data*, 5:180053, 2018. ISSN 2052-4463 (Electronic) 2052-4463 (Linking). doi: 10.1038/sdata.2018.53. URL https://www.ncbi.nlm.nih.gov/pubmed/29611842.

Jingzi Zhang, Mengkun Zhao, Chengquan Zhong, Jiakai Liu, Kailong Hu, and Xi Lin. Data-driven machine learning prediction of glass transition temperature and the glass-forming ability of metallic glasses. *Nanoscale*, 15:18511–18522, 2023. doi: 10.1039/D3NR04380K. URL http://dx.doi.org/10.1039/D3NR04380K.

Xiaoting Zhong, Brian Gallagher, Shusen Liu, Bhavya Kailkhura, Anna Hiszpanski, and T. Yong-Jin Han. Explainable machine learning in materials science. *npj Computational Materials*, 8(1), 2022. ISSN 2057-3960. doi: 10.1038/s41524-022-00884-7.

Ya Zhuo, Aria Mansouri Tehrani, and Jakoah Brgoch. Predicting the band gaps of inorganic solids by machine learning. *The Journal of Physical Chemistry Letters*, 9(7):1668–1673, 2018. doi: 10.1021/acs.jpclett.8b00124.

M. Ziatdinov, Y. Liu, K. Kelley, R. Vasudevan, and S. V. Kalinin. Bayesian active learning for scanning probe microscopy: From gaussian processes to hypothesis learning. *ACS Nano*, 16(9): 13492–13512, 2022. ISSN 1936-086X (Electronic) 1936-0851 (Linking). doi: 10.1021/acsnano.2c05303. URL https://www.ncbi.nlm.nih.gov/pubmed/36066996.

A APPENDIX

A.1 DATASETS

To assess the performance of PBNNs for AL on a variety of diverse datasets, we have selected two molecular and two materials datasets for benchmarking, and one molecular and one materials dataset containing both simulation and experimental data to investigate transfer learning (TL) from computed to experimental properties. Details, such as the dataset sizes and relevant references, regarding these datasets are provided in Tables A1 and A2. The FreeSolv, ESOL, and Steel fatigue (NIMS) datasets were used as published, while the Conductivity (HTEM) and Bandgap datasets are subsets of the published databases. Specifically, the Conductivity (HTEM) dataset utilized here is restricted to oxides containing Ni, Co, and Zn which have electrical conductivity values, and the Bandgap dataset is a random sample of 1000 non-metals from the intersection of the Materials Project bandgap dataset and the Matbench experimental bandgap dataset. We also used a noisy version of FreeSolv (Noisy-FreeSolv) for TL where experimental target values were corrupted by a zero-centered Gaussian noise with a standard deviation of one.

As far as the input features are concerned, we used standard RDKit (Contributors) physicochemical descriptors for the molecular datasets. For the steel fatigue dataset, the input features were chemical compositions, upstream processing details, and heat treatment conditions. For the electrical conductivity data, the input features were formed from oxide concentrations, deposition conditions, and processing parameters, such as power settings and gas flow rate. The input features for the Bandgap dataset were derived using the Magpie featurizer, which computes statistical descriptors from elemental properties and composition fractions (Ward et al., 2016).

Table A1: Datasets for Active Learning

Name	**Target property**	$N_{features}$	$N_{samples}$	**Reference**
FreeSolv	Hydration free energy	9	642	Mobley & Guthrie (2014)
ESOL	Aqueous solubility	9	1128	Delaney (2004)
Steel fatigue (NIMS)	Fatigue strength	25	437	Agrawal et al. (2014)
Conductivity (HTEM)	Electrical conductivity	12	1184	Zakutayev et al. (2018)

Table A2: Datasets for Transfer Learning

Name	**Target property**	$N_{features}$	$N_{samples}$	**Reference**
Noisy-FreeSolv	Hydration free energy	9	642	Mobley & Guthrie (2014)
Bandgap	Bandgap energy	132	1000	Jain et al. (2013); Zhuo et al. (2018)

A.2 ACTIVE LEARNING

In AL, the algorithm iteratively identifies points from a pool of unobserved data, within a pre-defined parameter space $\mathcal{X}_{\text{domain}} \subseteq \mathbb{R}^d$, that are expected to improve the model's performance in reaching some objective. Starting with an initial, usually small, training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, an initial PBNN is trained and predictions are made on all $x^* \in \mathcal{X}_{\text{domain}}$. The predictions that maximize a suitably selected acquisition function are then selected for measurement via an experiment, simulation, or human labeling. For the sake of benchmarking, we have chosen an acquisition function that simply maximizes the predictive uncertainty, *i.e.*, $x_{\text{next}} \leftarrow \arg\max_{x^* \in \mathcal{X}_{\text{domain}}} U(x^*)$, and only select a single x_{next} at each iteration. Note that here we naturally balance exploration between regions of model uncertainty and inherent complexity, as high aleatoric uncertainty often indicates areas requiring additional samples to better estimate noise distributions and capture underlying patterns. For further details regarding the AL algorithm, see Algorithm A1. Usually, this process is repeated until a desired goal is reached or an experimental budget is exhausted; here, we perform 200 exploration steps for all datasets. Lastly, we have selected initial training datasets by randomly sampling subsets of the total datasets containing 5% of the total number of data points. While this procedure results in differently sized initial training datasets, the trends observed are consistent across all datasets and corresponding sizes.

Algorithm A1 Active Learning

Require:
Parameter space $\mathcal{X}_{\text{domain}} \subseteq \mathbb{R}^d$
Number of initial measurements N
PBNN model architecture and parameters
Stopping criterion

1: Conduct N random measurements to create initial dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$
2: Train the PBNN on $\mathcal{D}$ using Algorithm 1
3: **repeat**
4: Compute PBNN's predictive uncertainty $U(x^*)$ for each $x^* \in \mathcal{X}_{\text{domain}}$
5: $x_{\text{next}} \leftarrow \arg\max_{x^* \in \mathcal{X}_{\text{domain}}} U(x^*)$
6: Perform measurement at x_{next} to obtain y_{next}
7: Update $\mathcal{D}$ by adding $(x_{\text{next}}, y_{\text{next}})$
8: Re-train the PBNN on updated $\mathcal{D}$ using Algorithm 1
9: **until** Stopping criterion is met

To assess AL performance, we computed several key metrics after each AL iteration. Our evaluation encompasses both prediction accuracy and uncertainty quantification.

Prediction accuracy was evaluated using the standard root mean square error (RMSE):

$$RMSE = \sqrt{\frac{\sum_i^M (y_i - \mu_i)^2}{M}}, \tag{A8}$$

where M is the size of the test set.

To assess the quality of the predictive uncertainties, we used two metrics, the negative log predictive density (NLPD) and the confidence interval coverage probability, which we refer to as coverage from this point forward. NLPD is given by the following equation:

$$\text{NLPD} = -\frac{1}{M}\sum_{i=1}^{M}\left[-\frac{1}{2}\log(2\pi U_i) - \frac{(y_i - \mu_i)^2}{2U_i}\right] \tag{A9}$$

NLPD assesses how well a model's predictive distributions align with observed data. A lower NLPD indicates that the model assigns higher probability density to true outcomes while maintaining well-calibrated uncertainty estimates. This metric is valuable for evaluating probabilistic models as it penalizes both overconfident incorrect predictions and underconfident correct ones.

Coverage is given by

$$Coverage = \frac{1}{M}\sum_{i}^{M} y_i \in \text{CI}(x_i), \tag{A10}$$

where $\text{CI}(x_i)$ is the confidence interval of test point x_i. Coverage measures the empirical reliability of a model's uncertainty estimates by calculating the proportion of true values that fall within the predicted confidence intervals (Kompa et al., 2021). In this work, all coverage values are computed for 95% confidence intervals.

A.3 Convergence Diagnostics

Figure A1 shows the distribution of R-hat values across PBNN (0, 4) parameters aggregated over all active learning steps for four different case studies: ESOL, FreeSolv, Steel fatigue, and HTEM datasets. All cases demonstrate good convergence characteristics, with the majority of parameters having R-hat values close to 1.0. The distributions exhibit a right-skewed pattern, which is expected in MCMC convergence diagnostics. There are, however, variations between datasets - particularly, the Steel fatigue case shows a wider spread of R-hat values, which correlates with more volatile NLPD values and slower Coverage convergence in early active learning steps. Nevertheless, most of the weights and biases fall within the the range $1.0 <$ R-hat < 1.1, which is traditionally considered to indicate good convergence.

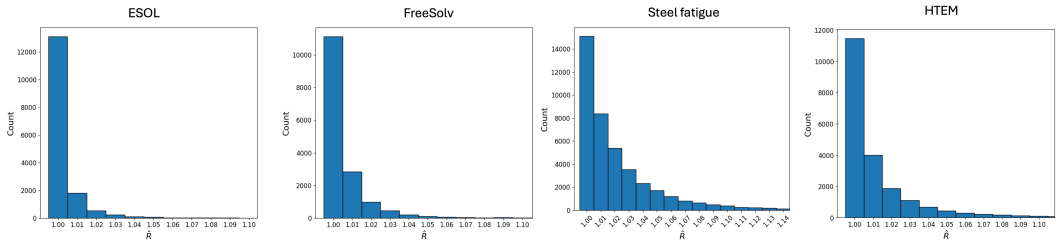

Figure A1: Gelman-Rubin ‘R-hat’ values over all active learning steps for ESOL, FreeSolv, Steel fatigue, and HTEM datasets.