SAND2024-07432J

Active Learning for SNAP Interatomic Potentials via Bayesian Predictive Uncertainty

Logan Williams[a], Khachik Sargsyan[a,*], Andrew Rohskopf[b], Habib N. Najm[a]

[a]*Sandia National Laboratories, Combustion Research Facility, Livermore, 94551, CA, USA*
[b]*Sandia National Laboratories, Center for Computing Research, Albuquerque, 87185, NM, USA*

Abstract

Bayesian inference with a simple Gaussian error model is used to efficiently compute prediction variances for energies, forces, and stresses in the linear SNAP interatomic potential. The prediction variance is shown to have a strong correlation with the absolute error over approximately 24 orders of magnitude. Using this prediction variance, an active learning algorithm is constructed to iteratively train a potential by selecting the structures with the most uncertain properties from a pool of candidate structures. The relative importance of the energy, force, and stress errors in the objective function is shown to have a strong impact upon the trajectory of their respective net error metrics when running the active learning algorithm. Batched training of different batch sizes is also tested against singular structure updates, and it is found that batches can be used to significantly reduce the number of retraining steps required with only minor impact on the active learning trajectory.

Keywords: interatomic potentials, Bayesian inference, uncertainty quantification, active learning

1. Introduction

The development of machine learned interatomic potentials (MLIAPs) has allowed for simulation with accuracy on par with *ab initio* methods such

*Corresponding author, ksargsy@sandia.gov

as Density Functional Theory (DFT) to be extended to calculations with billions of atoms due to the locality assumption in MLIAPs that gives $O(N)$ scaling [1]. MLIAPs typically rely on parametric or non-parametric functional forms that are optimized given a set of DFT data in a regression, or a supervised machine learning (ML), context. There is a large body of literature that develops and deploys such functional forms together with carefully crafted input characterizations, or descriptors, to aid the training process and conserve invariances and symmetries present in the system. These models include nonparametric forms, such as Gaussian approximation potentials (GAPs) [2], support vector machines [3], highly over-parameterized forms, such as neural networks [4], or linear expansions with judicious choice of input descriptors, such as moment tensor potentials [5], permutationally invariant polynomials [6], radial basis functions [7], or spectral neighbor analysis potential (SNAP) [8]. By training MLIAPs with flexible model forms on highly accurate DFT data of all relevant atomic environments, high accuracy of the model can be guaranteed. However, exhaustively collecting DFT data for all possible local atomic configurations is infeasible, particularly for multi-element systems.

Traditionally, training sets of DFT data were hand-crafted by subject matter experts. They would typically include the most stable crystal structure(s), expansions and contractions of said perfect crystals, surfaces, defect structures, various strained states of the crystals, and any small-scale examples of behaviors the model is being built to replicate, such as surface adsorption, melting, etc. Additional structures are often generated through some form of intentional randomness, such as elevated temperature molecular dynamics (MD) using a less-trained version of the model. This methodology can create very accurate models in the usage space it is designed for, but often extrapolates poorly to situations outside what it was designed for[9]. Additionally, expertise and substantial human time are both required. This calls for *active learning* (AL) methods which automatically select training sets based on pre-defined criteria [10]. AL has been widely employed in the ML community to minimize human-labeling efforts in classification tasks in computer vision. While underutilized for regression tasks in scientific ML in general, AL-based methods have been growing in popularity, in particular, when aiming to automate the selection of structures in training of MLIAPs. Many flavors of AL algorithms exist in this context, broadly categorized into three groups by their methods of defining structures to consider: pool-based, stream-based,

and synthesis-based; and into two groups by their method of evaluating the considered structures: descriptor diversity or output uncertainty. Diversity-based approaches select configurations based on a pre-defined notion of diversity or extrapolation [11]. For example, a hierarchical clustering algorithm was employed in [12] in order to diversify the selection of a training set. A kernel-based metric is employed to maximize diversity in GAP construction in [13] and [14]. A body of work relies on D-optimality to develop an extrapolation metric for linearly parameterized potentials [15, 16, 17, 18, 19], and its generalized version for non-linear potentials [20, 21]. The D-optimality metric is largely related to predictive uncertainty estimation of the potentials, which has become the more prominent approach to guide AL in interatomic potential construction. For example, MLIAPs based on Gaussian processes have a built-in predictive uncertainty estimation, which has been shown to enable AL for on-the-fly selection of configurations where the model predictions have large uncertainties [22, 23, 24, 25, 26, 27]. Statistics derived from ensembling approaches, albeit somewhat empirical, have proved to serve as accurate proxies for uncertainty estimation. Ensembles, or committees, of NNs have been employed in [28, 29, 30]. Similarly, query-by-committee is used to approximate prediction errors by committee prediction disagreement enabling efficient training data selection in [31, 32, 33]. Recently, Wilson *et al.* [34] combined uncertainty estimation via query-by-bagging and a distance-based similarity metric in a batch AL strategy for a potential construction based on kernel ridge regression potential. Arguably, Bayesian inference provides a more rigorous path toward uncertainty estimation [35, 36, 37, 38]. However, for non-linear models such as NNs, Bayesian predictions are intractable, although variational [39] and Gaussian mixture model approximations [40] have shown recent successes. For linearly parameterized models the potential training can be seamlessly lifted to a Bayesian realm, providing predictive uncertainty assessment at minimal extra cost.

In this work, we present an AL method for training the SNAP [8] interatomic potentials using Bayesian prediction variance. We augment the construction with K-means clustering to ensure a sufficiently diverse set of training data is selected. This study uses a pool-based approach, but the framework is fully compatible with stream-based AL workflows as well. By its Bayesian nature, it is also highly suitable to the forward propagation of uncertainty in production MD runs.

The paper is organized as follows. Section 2 describes the methodology, including the potential form, probabilistic training and the active learning

mechanism. Results are presented in Section 3, before we conclude in Section 4.

2. Methodology

2.1. Spectral Neighbor Analysis Potential

The Spectral Neighbor Analysis Potential (SNAP) is a class of interatomic potentials that are built as linear expansions with respect to bispectrum component descriptors. The bispectrum is a local neighbor density projected onto a basis of hyperspherical harmonics in four dimensions.[8, 41] Their calculation is performed as follows.

2.1.1. Bispectrum Components

The bispectrum components are functions of the atomic environment of each atom. The atomic density at radial location $\boldsymbol{r}$ from a central atom is written as a weighted sum of δ-functions in Cartesian space

$$\rho(\boldsymbol{r}) = \delta(\boldsymbol{r}) + \sum_{i:r_i<R_{\text{cut}}} f_c(r_i) w_i \delta(\boldsymbol{r} - \boldsymbol{r}_i) \tag{1}$$

where $\boldsymbol{r}_i$ is the radial location of atom i from the central atom, $r_i = \|\boldsymbol{r}_i\|$, $f_c(r_i)$ is a switching function that creates a smooth transition to zero as r_i nears the cutoff radius R_{cut}, w_i are dimensionless weights used to distinguish atoms of different elements.

Instead of expanding this density function (1) into an angular basis of spherical harmonics (θ, ϕ) and an orthonormal radial basis, the radial coordinate r is mapped onto a third angular coordinate following Bartók *et al.*[2]:

$$\theta_0 = \theta_0^{\max} \frac{r}{R_{\text{cut}}}.$$

This third angle allows for any set of points (r, θ, ϕ) in 3D space to be mapped onto an equivalent set of points (θ_0, θ, ϕ) that exist in 4D upon the unit-sphere (the 3-sphere).

The 4D hyperspherical harmonics, $U^j_{m,m'}(\theta_0, \theta, \phi)$ defined for $j = 0, \frac{1}{2}, 1...$ and $m, m' = -j, -j+1, ..., j-1, j$, form a natural basis for functions on the 3-sphere [42]. The atomic density around each central atom is thus expanded into the 4-D hyperspherical harmonics basis

$$\rho(\boldsymbol{r}) = \sum_{j=0,\frac{1}{2},...}^{\infty} \sum_{m=-j}^{j} \sum_{m'=-j}^{j} u^j_{m,m'} U^j_{m,m'}(\theta_0, \theta, \phi) \tag{2}$$

where $u^j_{m,m'}$, the complex valued expansion coefficients are given by the inner product of the neighbor density with the hyperspherical harmonics basis functions, $U^j_{m,m'}()$. As the neighbor density is a weighted sum of δ-functions, the expansion coefficients can be rewritten as a sum over discrete values of the corresponding basis function

$$u^j_{m,m'}(\boldsymbol{r}) = U^j_{m,m'}(0,0,0) + \sum_{i:r_i<R_{\text{cut}}} f_c(r_i) w_i U^j_{m,m'}(\theta_0, \theta, \phi). \tag{3}$$

Finally, from this notation, the bispectrum components are created as real-valued and rotationally invariant descriptors by taking the following triple products of the expansion coefficients:

$$b_{j_1,j_2,j}(\boldsymbol{r}) = \sum_{m_1,m_1'=-j_1}^{j_1} \sum_{m_2,m_2'=-j_2}^{j_2} \sum_{m,m'=-j}^{j} (u^j_{m,m'})^* H^{jmm'}_{\substack{j_1 m_1 m_1' \\ j_2 m_2 m_2'}} u^{j_1}_{m_1,m_1'}(\boldsymbol{r}) u^{j_2}_{m_2,m_2'}(\boldsymbol{r}) \tag{4}$$

where $*$ indicates complex conjugation and H are the Clebsch-Gordan coupling coefficients. These bispectrum components are symmetric in the indices j_1, j_2, j up to a normalization factor, giving $(j_{\max}+1)(j_{\max}+2)(j_{\max}+\frac{3}{2})/3$ unique descriptors for a given maximum value of j in the expansion. In their use as descriptors, these multi-indexed descriptors are flattened into a vector that will be denoted as $\mathbf{b}(\boldsymbol{r}) = (b_1(\boldsymbol{r}), \ldots, b_K(\boldsymbol{r}))$ in this paper, where K is the number of bispectrum components determined by the truncation of the j, j_1, j_2 indices.

2.1.2. SNAP Potential Form

Next, let $\mathbf{r} = (x, y, z)$ be the Cartesian coordinate vector, with the location of atom i denoted as $\mathbf{r}_i$. Further, aggregating all Cartesian coordinates that define an N-atom structure, we denote $\mathbf{r}^N = (\mathbf{r}_1, \ldots, \mathbf{r}_N)$. The energy of a crystal structure composed of N atoms is represented as a linear model with respect to bispectrum components

$$E_{SNAP}(\mathbf{r}^N) = N\beta_0 + \boldsymbol{\beta} \cdot \sum_{i=1}^{N} \mathbf{b}^i(\mathbf{r}^N) \tag{5}$$

where $\mathbf{b}^i(\mathbf{r}^N)$ is the flattened K-vector of bispectrum components for the i-th atom in the structure with requisite transformations to atom-centric coordinates, $\boldsymbol{\beta}$ is the vector of coefficients for the linear model, and β_0 is

an offset constant that can be fit by the model, specified, or ignored. Note that both $\boldsymbol{\beta}$ and β_0 are dependent on atom type: in systems with n types of atoms, there will be n sets of coefficients $(\beta_0, \boldsymbol{\beta})$.

The force on each atom is given by the negative gradient of the energy

$$\boldsymbol{f}^j_{SNAP}(\mathbf{r}^N) = -\nabla_j E_{SNAP}(\mathbf{r}^N) = -\boldsymbol{\beta} \cdot \sum_{i=1}^{N} \frac{\partial \mathbf{b}^i(\mathbf{r}^N)}{\partial \mathbf{r}_j} \tag{6}$$

for $j = 1, 2, \ldots, N$. Next, the stress tensor is computed as

$$\mathbf{W}_{SNAP}(\mathbf{r}^N) = -\sum_{j=1}^{N} \mathbf{r}_j \otimes \nabla_j E_{SNAP}(\mathbf{r}^N) = -\boldsymbol{\beta} \cdot \sum_{j=1}^{N} \mathbf{r}_j \otimes \sum_{i=1}^{N} \frac{\partial \mathbf{b}^i(\mathbf{r}^N)}{\partial \mathbf{r}_j}, \tag{7}$$

where $\otimes$ is the Cartesian outer product operator. The stress tensor for systems of interest is symmetric, thus including 6 independent scalar components.

With these definitions, the overall SNAP model can be written as a linear model $\boldsymbol{y}(\mathbf{r}^N) = \mathbf{B}(\mathbf{r}^N)\boldsymbol{c}$ where the coefficient vector has length $K + 1$ and is defined as $\boldsymbol{c} = (\beta_0, \boldsymbol{\beta})^T$. The data vector has $3N + 7$ elements

$$\boldsymbol{y}(\mathbf{r}^N) = (E_{SNAP}(\mathbf{r}^N), \mathbf{f}_{SNAP}(\mathbf{r}^N), \mathbf{W}_{SNAP}(\mathbf{r}^N)), \tag{8}$$

while the matrix $\mathbf{B}$ has dimensions $(3N + 7) \times (K + 1)$ with rows directly following Eqs (5), (6) and (7) corresponding to energy, force and stress components. The energy row is defined as $(N, \sum_{i=1}^{N} \mathbf{b}^i)$, each triplet of the $3N$ force rows are defined as $(0, -\sum_{i=1}^{N} \partial \mathbf{b}^i / \partial \mathbf{r}_j)$, and the 6 stress rows are defined based on $(0, -\sum_{j=1}^{N} \mathbf{r}_j \otimes \sum_{i=1}^{N} \partial \mathbf{b}^i / \partial \mathbf{r}_j)$.

2.2. SNAP Training

The goal of fitting SNAP potentials is to find the coefficient vector $\boldsymbol{c}$ given a set of T *training* configurations with corresponding energy/force/stress values $\left(\mathbf{r}^{(t)}, \boldsymbol{y}(\mathbf{r}^{(t)})\right)$, for $t = 1, \ldots, T$, each of which entails $3N + 7$ rows of the $\mathbf{B}(\mathbf{r}^{(t)})$ matrix and $3N + 7$ elements of the data vector $\boldsymbol{y}$ obtained by DFT computations. From here onward, we drop the superscript from $\mathbf{r}^N$ for convenience, letting $\mathbf{r} \equiv \mathbf{r}^N$. The objective is to find coefficients $\boldsymbol{c}$ that minimize the training loss function which is a least squared error defined as

$$L(\boldsymbol{c}) = \sum_{t=1}^{T} ||\mathbf{B}(\mathbf{r}^{(t)})\boldsymbol{c} - \boldsymbol{y}(\mathbf{r}^{(t)})||^2. \tag{9}$$

With a further slight abuse of notation, we denote the overall matrix of bispectrum components evaluated at all T training configurations as $\mathbf{B}$, which has a size of $(3N+7)T \times (K+1)$. Similarly the overall data vector is $\boldsymbol{y}$ and is of size $M = (3N+7)T$. Because the model is linear with respect to the coefficients, one can write a closed-form analytical solution to the training problem as

$$\boldsymbol{c}^* = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\boldsymbol{y}. \tag{10}$$

Typically, the final, fitted SNAP model is added to a simple reference value. That is, the $\boldsymbol{y}$ vector that the model is fit to is not the raw DFT data, but the difference between the raw DFT data and the simple, reference model prediction value $\boldsymbol{y} = \boldsymbol{y}^{DFT} - \boldsymbol{y}^{ref}$. Furthermore, fitting SNAP is typically done using a weighting scheme that involves multiplying all the rows of the $\mathbf{B}$ matrix (i.e. bispectrum components) and the data vector $\boldsymbol{y}$ element-wise by a weight vector, $\boldsymbol{w}$ of the same size as $\boldsymbol{y}$. For ease of presentation, we will keep the notation $\boldsymbol{y}$ and $\mathbf{B}$, absorbing the weight multiplication into them without loss of generality.

Next, in order to inform active learning strategies of selecting training configurations with highest model uncertainties, we will generalize the deterministic training of SNAP using Bayesian methods to arrive at a probabilistic description of coefficients $\boldsymbol{c}$ and consequently endow the associated predictions $\mathbf{B}\boldsymbol{c}$ with uncertainty information.

2.3. Bayesian Regression for SNAP Training

Bayesian inference provides a consistent framework for incorporating uncertainties in model parameter estimates, relying on Bayes' rule, which in this context is stated as [43]

$$p(\boldsymbol{c}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{c})p(\boldsymbol{c})}{p(\boldsymbol{y})}. \tag{11}$$

In this equation a prior probability density function (PDF) $p(\boldsymbol{c})$ is updated in light of data $\boldsymbol{y}$ to arrive at the object of interest, the *posterior* PDF $p(\boldsymbol{c}|\boldsymbol{y})$. The denominator $p(\boldsymbol{y})$ is called the evidence, and plays the role of a normalizing factor. The key to Bayes' rule is the *likelihood* function $p(\boldsymbol{y}|\boldsymbol{c})$ which incorporates assumptions about the noise in the data $\boldsymbol{y}$ and the discrepancy between it and the model prediction implied by coefficient vector $\boldsymbol{c}$.

In the present context, and to enable analytical tractability, we make an uninformative improper prior assumption $p(\boldsymbol{c}) \equiv \text{const}$, and an independent identically distributed (*iid*) Gaussian noise assumption for the likelihood

$$p(\boldsymbol{y}|\boldsymbol{c}) = \frac{1}{\sqrt{2\pi}^M \sigma^M} \exp\left(-\frac{||\mathbf{B}\boldsymbol{c} - \boldsymbol{y}||^2}{2\sigma^2}\right), \tag{12}$$

where σ is the assumed data noise magnitude in the Gaussian data generating model $y_m = (\mathbf{B}\boldsymbol{c})_m + \sigma\epsilon_m$, where $\epsilon_m \sim \mathcal{N}(0,1)$ and $m = 1, \ldots, M$. In our formulation, we incorporate σ as an object of inference together with coefficients $\boldsymbol{c}$, and work with a slightly altered Bayes' rule

$$p(\boldsymbol{c}, \sigma^2|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{c}, \sigma^2)p(\boldsymbol{c})p(\sigma^2), \tag{13}$$

where the prior PDF on σ^2 of the form $p(\sigma^2) = 1/\sigma^2$ is a commonly used Jeffrey's prior, corresponding to a uniform prior assumption on $\log\sigma$. This formulation allows for a closed-form analytical expression for the posterior PDF of the coefficients after integrating with respect to σ^2 [44]. Namely, one obtains a Student-t distribution of the form $\boldsymbol{c} \sim \text{St-t}(\boldsymbol{c}^*, \sigma_*^2(\mathbf{B}^T\mathbf{B})^{-1}, M-K-1)$ with mean $\boldsymbol{c}^*$ that coincides with the deterministic answer from Eq. (10), the optimal (i.e. maximum posterior) value of data noise magnitude

$$\sigma_*^2 = \frac{||\mathbf{B}\boldsymbol{c}^* - \boldsymbol{y}||^2}{M-K-1}, \tag{14}$$

and the degrees-of-freedom parameter $M-K-1$ being the number of data points minus the number of basis terms. In the typical limit of $M \gg K$, the Student-t distribution is very closely approximated by a multivariate normal (MVN) distribution $\boldsymbol{c} \sim \text{MVN}(\boldsymbol{c}^*, \boldsymbol{\Sigma})$, where we introduce a notation for the covariance matrix $\boldsymbol{\Sigma} = \sigma_*^2(\mathbf{B}^T\mathbf{B})^{-1}$. Now, since the posterior PDF of the coefficients $\boldsymbol{c}$ is an MVN distribution and the SNAP model is a linear function acting on those coefficients, one can deduce that predictions on any set of training or testing configurations are MVN, too. Namely,

$$\mathbf{B}_{\text{new}}\boldsymbol{c} \sim \text{MVN}(\mathbf{B}_{\text{new}}\boldsymbol{c}^*, \mathbf{B}_{\text{new}}\Sigma\mathbf{B}_{\text{new}}^T), \tag{15}$$

where $\mathbf{B}_{\text{new}} = \mathbf{B}(\mathbf{r}_{\text{new}})$ is a new matrix constructed similarly to the original training matrix $\mathbf{B}$, but with a new set of configurations $\mathbf{r}_{\text{new}}$. Note that for active learning described in the next section, we are interested in prediction variance only, hence the matrix product in the covariance of Eq. (15) is not evaluated fully: instead we compute its diagonal without incurring significant computational costs.

2.4. Active Learning

Overall, the SNAP training, i.e. computation of mean prediction and the associated uncertainty estimation, as described in Sections 2.2 and 2.3, is relatively inexpensive, the major bottleneck being the DFT evaluations that form the training set. For this reason, it is crucially important to implement active learning strategies that allow identifying structures to evaluate DFT for in a manner that optimally aids the SNAP construction. Armed with the Bayesian uncertainty estimates, we develop a pool-based active learning (AL) method to automatically select structures for which new DFT data needs to be computed to augment the training set. The procedure starts with a large pool of "unlabeled" structures, *i.e.* structures for which DFT solutions have not been computed and/or have not been used for training. We choose a sample of structures to provide the initial training data set, label it by computing its DFT solutions, and use it to train a SNAP model. Structures that have been labeled and used to train the model are subsequently excluded from the AL pool. Using the trained model we then estimate SNAP predictive uncertainties for all structures in the pool and, based on various *acquisition functions*, decide which subset of structures should be added to the training set. Further, in order to enhance diversity in the next batch of structures, we apply k-means clustering first, and pick no more than one structure per cluster. With the updated training set, we re-train the SNAP model, and repeat the cycle. The choice of the initial pool of unlabeled structures, as well as that of the initial training data set, particularly their diversity and sizes, are both important, but are not considered here. We leave these explorations for future work. Here, we simply start with a pre-selected diverse pool.

Next, we describe the choices we make in this work for the two ingredients of the AL algorithm, acquisition function and diversity choice, as well as the metric we employ to account for the associated cost of the overall algorithm.

Acquisition Function Choices. As detailed in Section 2, for every structure in the pool, there exists one system energy, $3N_i$ forces, and 6 stress values to be used as training data, where N_i is the number of atoms in the i-th structure. Acquisition functions will rely on prediction variances as detailed in Section 2.3. However, selection of *how* to combine prediction variances across various structures and the usage of the energy, force and stress uncertainties leaves some room for investigation. Below we highlight several options in this regard tested in this work.

Let us denote predictive variances, i.e. the appropriate diagonal elements of the Bayesian predictive covariance from Eq. (15), as follows. We denote the predictive variance for the energy of the i-th structure as V_{E_i}, the j-th element of the associated force prediction variance as $V_{F_i}^j$, for $j = 1, \ldots, 3N_i$, and the j-th stress variance as $V_{S_i}^k$, for $k = 1, \ldots, 6$.

Our AL acquisition functions target structure selection, and hence are aggregated summary metrics for each structure. Higher granularity, such as per-atom energy values, is difficult to obtain as the DFT data is on the total energy and lacks the additional per-atom detail. However, we have tried energy-only per-atom max or average acquisition functions with results similar to the `E_only` case, hence we omitted them for brevity. Aside from the random selection of structures (passive learning, denoted as `random` in figures) as a validation, we test here six such acquisition functions calculated from the predictive uncertainties. These are listed below, where we introduce a nomenclature (*e.g.* `E_only`) for the different cases, and denote selected structures with the index i_Q. Note that the structure weights are absorbed in the notation for energy, force and stress values, and hence they propagate similarly into the variance estimates. With that, we have left the acquisition functions as described below without additional weighting.

- `E_only`: $i_Q = \arg\max_i V_{E_i}$, selection based on the highest energy uncertainty,
- `F_only_average`: $i_Q = \arg\max_i \frac{1}{3N_i} \sum_{j=1}^{3N_i} V_{F_i}^j$, selection based on highest average force uncertainty in each structure,
- `F_only_max`: $i_Q = \arg\max_i \left(\max_j V_{F_i}^j\right)$, selection based on highest individual force uncertainty in each structure,
- `average`: $i_Q = \arg\max_i \frac{1}{3N_i+7} \left(V_{E_i} + \sum_{j=1}^{3N_i} V_{F_i}^j + \sum_{k=1}^{6} V_{S_i}^k\right)$, selection based on highest average of all energy, force, and stress uncertainties in each structure,
- `max`: $i_Q = \arg\max_i \left(\max(V_{E_i}, \max_j V_{F_i}^j, \max_k V_{S_i}^k)\right)$, selection based on highest individual uncertainty (energy, force, or stress) in each structure,
- `weighted_sum`: $i_Q = \arg\max_i \frac{1}{N_i^3} \left(V_{E_i} + \sum_{j=1}^{3N_i} V_{F_i}^j + \sum_{k=1}^{6} V_{S_i}^k\right)$, selection based on sum of all energy, force, and stress uncertainties in struc-

ture divided by the number of atoms cubed (to weigh according to approximate DFT cost).

Diversity Choice. While the cost of generating the DFT training data is typically the most computationally intensive step of potential training, the cost of refitting the model and re-evaluating the uncertainties is the second highest cost. Hence, it is natural to select a batch of structures per active learning step instead of adding one structure at a time. The top L structures according to the acquisition functions are chosen for the next batch to be added to the training set. Furthermore, in order to avoid adding too many similar structures at once, we employ a clustering algorithm to diversify the top-L choice. Namely, we divide the entire unlabeled dataset into 20 clusters using k-means clustering, where we use the (unscaled) system energy bispectrum descriptors to define the coordinate space with the associated Euclidean distance metric. The clusters are fixed and are not recalculated when structures are removed from the unlabeled pool and added to the training set. Our clustering-enhanced AL method enforces no more than one structure selected per cluster. This is always feasible since, in all examples in this work, the number of new structures is smaller than the pre-defined number of clusters, 20. In other words, we skip any additional structures from a cluster that has already had a structure selected in the current AL step.

Cost. In our studies of various AL acquisition function and diversity selection choices, we will use an empirical DFT expense estimate according to cubic scaling, i.e. for all the structures in the training sets, the 'expense' of the DFT evaluation of that structure is approximated as N_i^3.

3. Results

We rely on the FitSNAP library [45] available at `https://github.com/FitSNAP/FitSNAP` to compute SNAP bispectrum components, extract the example datasets, and perform SNAP training via Bayesian regression as described in Section 2.

3.1. Datasets

All three datasets are publicly available on the FitSNAP GitHub repository. Each dataset was split into a small initial training set, a standalone test set, and a remainder to be used as the pool, referred to as 'unlabeled' because the algorithm is assumed to not be aware of the labels.

Ta dataset. The tantalum dataset is small, containing 363 crystal structures. It includes perfect, pressurized, distorted, and atomically displaced versions of the FCC, BCC, and A15 crystal structures, as well as 110 and 112 stacking faults, 100, 110, 111, and 112 surfaces, and liquid structures from MD. The data is fitted using a bispectrum set with $2j_{\max} = 6$. The initial training dataset in this case is based on a random sampling created by shuffling the dataset and selecting the top 12, so it is should be roughly distributed like the overall pool.

InP dataset. The indium phosphide dataset is medium-sized, containing 1894 crystal structures. It consists of crystals in various states of strain and a variety of simple defect structures, including substitutions, vacancies, and interstitials. The data is fitted using a bispectrum set with $2j_{\max} = 6$ for both In and P. Unlike the small illustrative Ta dataset, this, as well as the next, WBe dataset are much closer, both in size and in importance, to representative datasets of practical use. The initial training set is selected to be a small number of perfect crystals in random states of uniform compression or expansion, achieving a fairly minimal information start of active learning.

WBe dataset. The tungsten-beryllium dataset is moderately large, containing 25422 crystal structures. It includes a variety of types of crystal structures including perfect crystals at different degrees of compressive strain and shear deformation, defects, surfaces, ab initio MD structures at 300K and 1000K, liquid structures, and stacking faults. The data is fitted using a bispectrum set with $2j_{\max} = 8$ for both W and Be. Similar to the InP case, the initial training set is selected to be a small number of perfect crystals in random states of uniform compression or expansion.

3.2. SNAP Predictive Uncertainty

The Bayesian prediction variance, derived as the diagonal of the covariance in Eq. (15), produces an imperfect but still very strong correlation with the absolute error across many orders of magnitude. As shown in Figure 1, the square root of the prediction variance matches well with the absolute error of each entry in the data pool across the $\sim$24 orders of magnitude spanned by both the Ta and InP datasets. The plots in Figure 1 were created after fitting the SNAP model using the starting training set, before any AL steps have occurred. Details of the study, including the initial training set sizes are given in the following sections where the full AL algorithm is

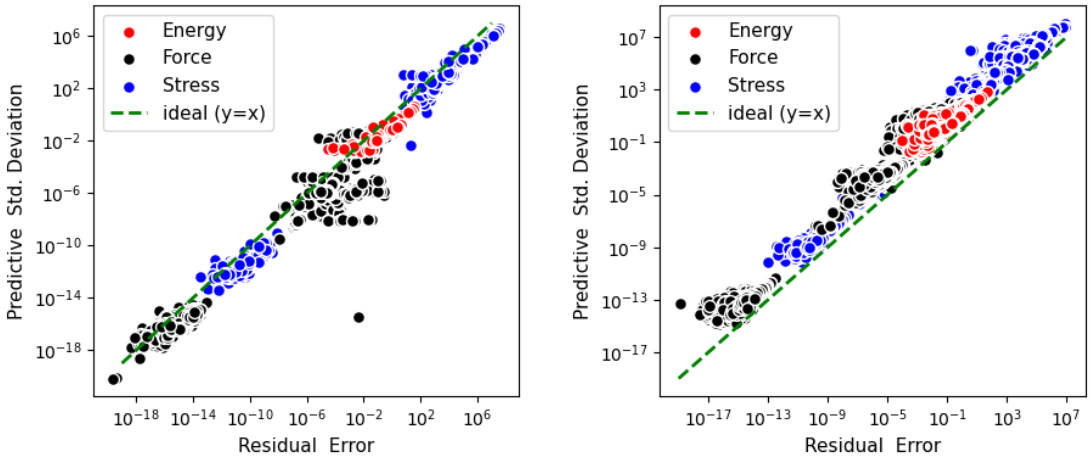

Figure 1: Plot of the square root of the prediction variance vs the absolute error of every value (energy, force, and stress) in the data pools of the (left) Ta and (right) InP datasets after training the first SNAP potential with their respective starting training data.

investigated. Figure 1 demonstrates error and predictive variance correlation for the structures in the active learning pool, i.e. data from training set of structures is excluded from the plot. The very small errors are force points as many atoms are in equilibrium for the structures considered. The strong correlation between predictive standard deviation and the true residual error is observed – fundamentally this is a crucial observation that the success of the presented AL algorithm relies on. Note that while the ideal $y = x$ is plotted, for the AL method to be successful one only needs a correlation – and not an exact match – between the predictive standard deviation and the actual residual. Also, we have observed that this correlation remains strong throughout all AL steps.

3.3. Acquisition Function Study

Ta dataset. For this study, 12 randomly selected structures were separated out to form the initial training set and 12 other randomly selected structures were set aside as an independent test set that would never be seen by the AL algorithm or the model fitting. SNAP hyperparameters and training group weights were left identical to their defaults from the FitSNAP repository.

This dataset exhibits predictable trends for the various acquisition func-

tions. As shown in Figure 2, using the `E_only` strategy produces a rapid early drop in the error on the energies of the test set. The AL energy test errors fall much faster than the random strategy with increasing AL-governed training data size. This AL test error decay rate slows down as more data points are added to the training set, which is very much a reflection of the finite size of the data pool. As we approach the upper limit of available training data to choose from in the pool, there is gradually less advantage from AL, given the limited choices available. Eventually, in this and all other cases in this work, given the finite pool, the AL advantage disappears when using the full available training data, so that the errors from the random and AL strategies are identical in that limit. Considering the *training* errors in Figure 2, those

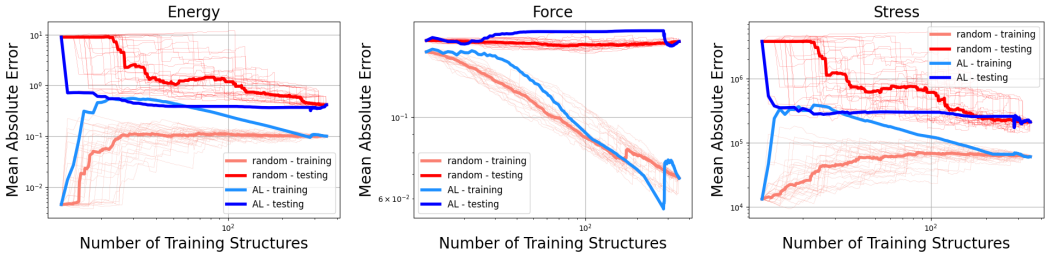

Figure 2: The testing (dark) and training (light) mean absolute errors on the Ta dataset for the AL algorithm (blue) using only energy uncertainties in the acquisition function (`E_only`) and a random selection algorithm (red). The thin lines show multiple runs of random selection, with the thick red line showing the median for all the runs at each point along the x-axis.

rise more rapidly early on with AL than they do for random selection, as the linear model cannot perfectly fit the dataset and the actively-selected crystal structures have (by construction) higher variance in energy than the randomly selected ones. As we go to larger training data sizes, the AL training error decays again, to arrive at the same level as the random strategy for the full data set in the pool as expected. The stress follows very similar curves to the energy, both in terms of testing and training errors. The crystal structures in this dataset with the highest energy variance are also the ones with the highest stress variance: perfect crystals with very large, uniform tensile or compressive strain. Meanwhile, compared to random selection, the force errors are worse with AL using the `E_only` strategy. The difference is slight, however, and not consistent at all data sizes. This suggests that those structures with high energy error estimate that are selected early on with the

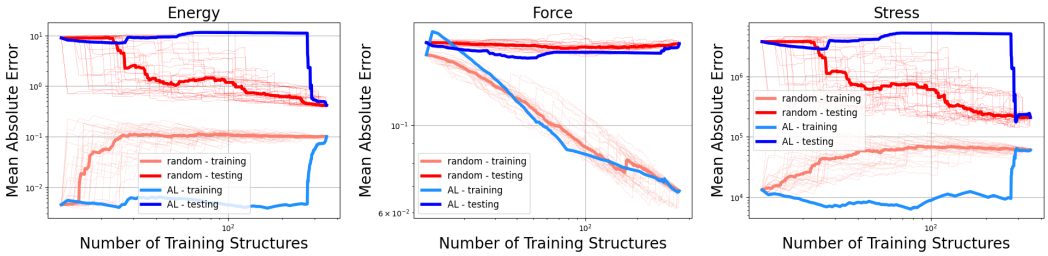

Figure 3: The testing (dark) and training (light) mean absolute errors on the Ta dataset for the AL algorithm (blue) using the average of only force uncertainties in the acquisition function (`F_only_average`) and a random selection algorithm (red). The thin lines show multiple runs of random selection, with the thick red line showing the median for all the runs at each point along the x-axis.

`E_only` strategy, while worthwhile for reducing energy and stress test errors, are not necessarily that useful for reducing force test errors. As a result, the `E_only` AL strategy is nearly equivalent to random selection as far as force test error reduction is concerned.

Active learning results using the `F_only_average` strategy are shown in Figure 3. Here we see that the opposite trends as compared to the `E_only` results in Figure 2, as might be expected. In fact, the situation is rather decidedly opposite. Force test set errors are indeed slightly improved compared to random selection, while energy and stress test set errors are significantly worse. Thus, AL data selection using `F_only_average` is not advisable in the present context.

Figure 4 shows the test set root mean square errors (RMSE) for all 6 acquisition functions and the random selection comparison. Starting from this figure, we rescale the x-axis to indicate the approximate computational cost for the DFT as described at the end of Section 2. The `max`, `average`, and `weighted_sum` acquisition functions all perform similarly, with fairly similar performance to the energy only acquisition function as well. The two acquisition functions using only force uncertainties behave very similarly to each other as well. The errors on the forces change only slightly ($< 10\%$), while the `max`, `average`, and `weighted_sum` acquisition functions all converged the errors on the energies and stresses extremely quickly compared to random selection. Note the sudden drop of the force-focused acquisition functions due to addition of structures with negligible force. Those are typically Equation of State (EOS) perfect crystal structures that are important for energy

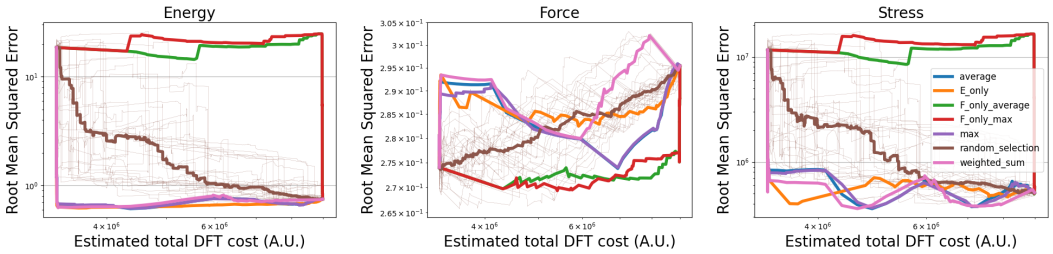

Figure 4: Test set root mean square errors (RMSE) on the Ta dataset for the six AL acquisition strategies and a random selection algorithm. The x-axis is an arbitrary units representation of the approximate DFT computational cost, calculated by summing empirical cost (N_i^3) of each structure in the training data. The thin lines show multiple runs of the random selection, with its thick line showing the median for all the runs at each point along the x-axis. The color legend applies to all plots.

prediction. Since these are very small structures, the x-axis is 'compressed' due to the low DFT cost. Generally, the force RMSE does not vary much: it looks to be saturated, and the AL algorithm largely improves the energy and stress predictions only.

InP dataset. For this study, 6 strained perfect crystal structures were randomly selected from the EOS datagroup to form the initial training set, and 70 other structures were randomly selected from the entire dataset as an independent test set that was never seen by the AL algorithm or the model fitting. All training group weights were set to 1.0 for energy and forces, and 10^{-12} for stresses, thus effectively ignoring stress data. In all our cases, including the Ta case where the stress is actually included in the training set, the stress fit is closely correlated with the energy fit, suggesting that the stress inclusion in the training set is not necessarily adding information. The original paper on the InP potential [46] also excluded the stress information almost entirely for similar reasons.The acquisition functions tested on this dataset also ignored stresses, but were otherwise unchanged. Figure 5 shows the test set RMSE for all 6 acquisition functions and the random selection comparison, with the x-axis as the approximated computational cost of the training set. Overall, the dataset displays only modest overall reduction in test set errors as training data is added, with the majority of the improvement very early in the process. Contrary to the Ta dataset behavior, the force errors display a continuous convergence throughout the addition of

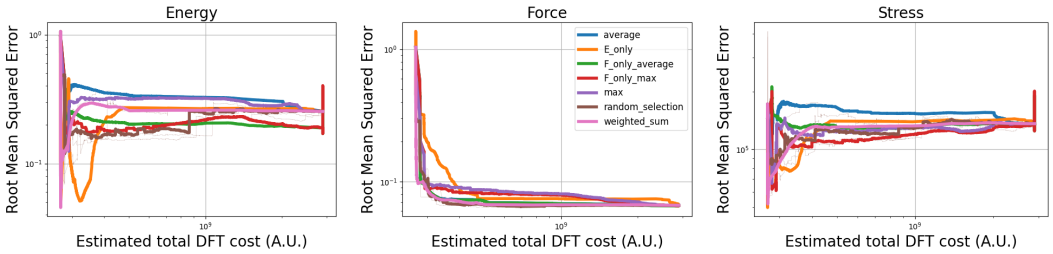

Figure 5: Test set RMSE on the InP dataset for the six AL acquisition strategies and a random selection algorithm. The x-axis is an arbitrary units representation of the approximate DFT computational cost, calculated by summing empirical cost (N_i^3) of each structure in the training data. The thin lines show multiple runs of the random selection, with its thick line showing the median for all the runs at each point along the x-axis. The color legend applies to all plots.

training data, while the energy and stress only benefit from select structures at the start. To better visualize the early convergence performance of the various algorithms, Figure 6 shows just the first 2% of the x-axis for each plot in Figure 5. The `weighted_sum` algorithm, which uses the N_i^3 scaling in its acquisition function, achieves the overall fastest initial reduction in test set errors before converging towards the full dataset values. In this study, we started with a limited training set (only a few EOS structures instead of a random sampling across the total dataset) highlighting how little data is needed in these models to reduce the errors and enter into a convergent RMSE behavior.

WBe dataset. For this study, 10 total strained perfect crystal structures were randomly selected from the EOS datagroups (2 each) to form the initial training set and 177 other structures were randomly selected from the entire dataset as an independent test set that was never seen by the AL algorithm or the model fitting. Similar to the previous dataset, all training group weights were set to 1.0 for energy and forces, and 10^{-12} for stresses, and the acquisition functions used for this dataset also were set to ignore stresses. As noted in the InP discussion, we observed that the addition of stress data is not too informative. The original paper on the WBe potential [47] also excluded the stress information almost entirely for similar reasons.

Figure 7 shows the test set RMSE for all 6 acquisition functions and the random selection comparison, with the x-axis as the approximated computational cost of the training set. For this dataset, to keep the computational

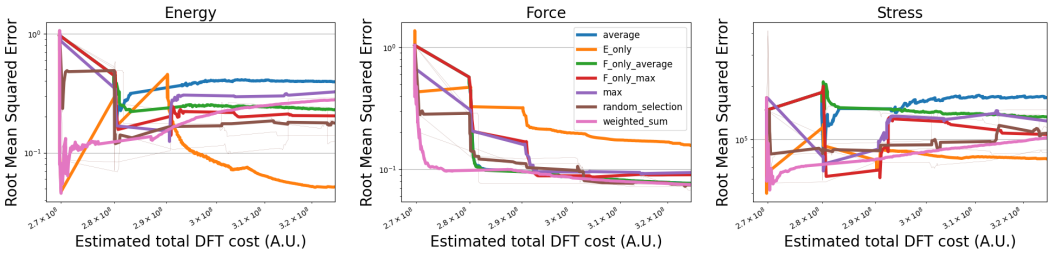

Figure 6: Test set RMSE on the InP dataset for the six AL acquisition strategies and a random selection algorithm. The x-axis is an arbitrary units representation of the approximate DFT computational cost, calculated by summing empirical cost (N_i^3) of each structure in the training data. Only the first 2% of the entire x-axis is shown. The thin lines show multiple runs of the random selection, with its thick line showing the median for all the runs at each point along the x-axis. The color legend applies to all plots.

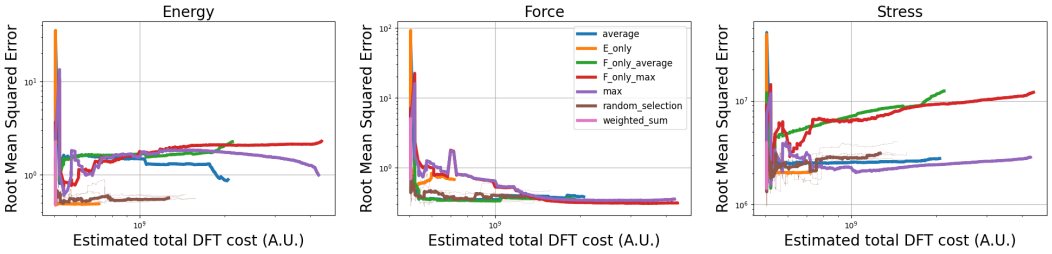

Figure 7: Test set RMSE on the WBe dataset for the six AL acquisition strategies and a random selection algorithm. The x-axis is an arbitrary units representation of the approximate DFT computational cost, calculated by summing empirical cost (N_i^3) of each structure in the training data. The thin lines show multiple runs of the random selection, with its thick line showing the median for all the runs at each point along the x-axis. The color legend applies to all plots.

cost at practically relevant values, only the first 5000 steps of the AL algorithm were performed for each acquisition function. Thus all methods do not fully converge. The `weighted_sum` in particular covers very little of the x-axis, biased as it is towards structures with low DFT cost. The `E_only` algorithm continues to perform well for both energy and stress error reduction. The `F_only_average` and `average` algorithms do well for force error reduction throughout, while `F_only_max` and `max` have an initial poor force convergence but overtake the other methods after many iterations.

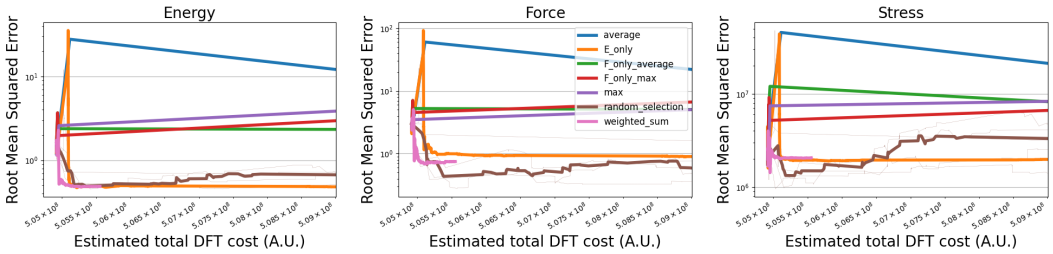

Figure 8: Test set RMSE on the WBe dataset for the six AL acquisition strategies and a random selection algorithm. The x-axis is an arbitrary units representation of the approximate DFT computational cost, calculated by summing empirical cost (N_i^3) of each structure in the training data. Only the first 0.1% of the entire x-axis is shown. The thin lines show multiple runs of the random selection, with its thick line showing the median for all the runs at each point along the x-axis. The color legend applies to all plots. Straight lines appear due to zooming in when only one point estimate is visible in the window.

Finally, Figure 8 shows the first 0.1% of the x-axis for each plot in Figure 7. In all cases, the `weighted_sum` is able to achieve rapid reduction in test set errors through the selection of low DFT cost structures. Random selection, and to a lesser extent `E_only` acquisition functions result in comparably good error reduction on this very short time scale. The rest of acquisition functions select large structures that are expensive to calculate in DFT resulting in low resolution of the corresponding curves at this zoomed-in scale.

3.4. Diversity Choice Study

Here, we investigate the performance of the `weighted_sum` acquisition selection algorithm when selecting multiple structures at a time, both unclustered (simply adding the structures with the highest X objective function evaluations) and utilizing a k-means clustering method as described in Section 2. Figure 9 shows the clusters identified by the k-means clustering algorithm on the energy bispectrum descriptors for the Ta dataset. The clusters primarily split according to the atomic volume of the structure. Structures with volume overlaps are sometimes clustered separately, such as the lower energy surface structures around 35 Å^3/atom, or the separation between the slightly higher energy FCC structures and the lower energy BCC and A15 structures between 10 and 13 Å^3/atom. Across most of the atomic volume range, though, the highly strained FCC, BCC, and A15 structures tend to

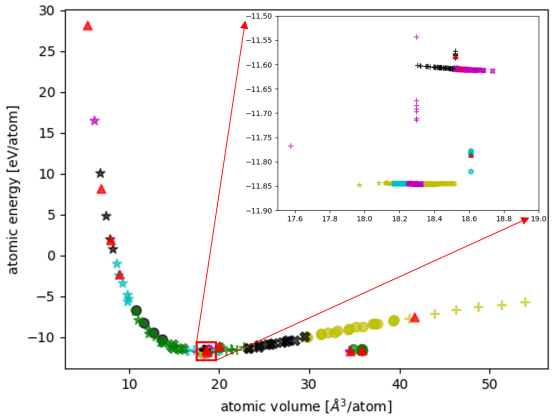

Figure 9: Plot of the atomic volume vs atomic energy for each structure in the Ta dataset's unlabeled pool, with different combinations of marker style and color representing different clusters found by the k-means clustering algorithm. The upwards pointing triangles (red) represent the independent test set.

cluster together according to their atomic volume. The one strong exception is the single liquid structure in the unlabeled pool which clusters with the 112 surface structures (atomic volume ≈ 35.9, atomic energy ≈ -11.5) despite its very different volume at an atomic volume ≈ 20.0 and an atomic energy at ≈ -11.2. Figure 10 shows the test set RMSE on the Ta dataset for both random selection and the weighted sum objective function using batch sizes of 1, 2, 5, and 10 for both unclustered and clustered batch addition. All batching and clustering settings tested behaved largely similarly. Increasing the batch size increased the deviation from the non-batched algorithm for both the clustered and non-clustered methods. We note that in the non-clustered, naïve batching method (green, red and purple curves) the batch size affects the convergence 'path' only slightly. The clustered batching, on the other hand, achieves diversity, and is more stable, in the sense that the convergence depends on the batch size less (brown, pink and gray curves). These observations held also for the InP and WBe datasets, the results being skipped for brevity of presentation.

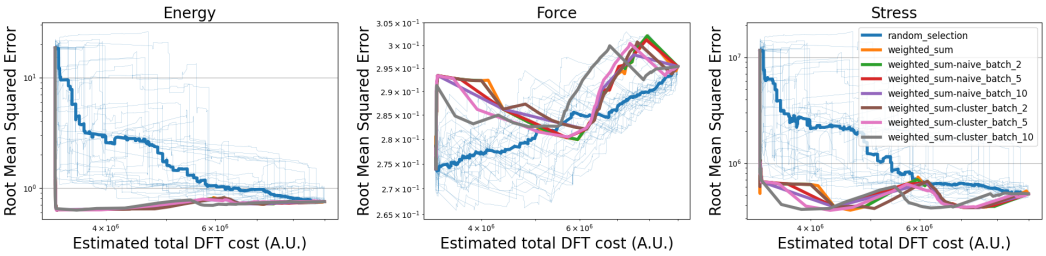

Figure 10: Test set RMSE on the Ta dataset using random selection and the weighted sum algorithm with batch sizes of 1, 2, 5, and 10. Both unclustered batching and clustered batching are shown, with clustered batching creating 20 clusters on the system energy descriptors using k-means clustering, allowing 1 structure to be chosen per batch per iteration. The x-axis is an arbitrary units representation of the approximate DFT computational cost, calculated by summing empirical cost (N_i^3) of each structure in the training data. The thin lines show multiple runs of the random selection, with its thick line showing the median for all the runs at each point along the x-axis. The color legend applies to all plots.

4. Conclusions

Bayesian inference using a simple Gaussian error model allows for robustly obtaining prediction variances in linearly parameterized models such as the SNAP IAP with relative little computational expense. These prediction variances are shown to have a strong correlation with the absolute error over 24 orders of magnitude. Active learning workflows can be built upon these prediction variances, with many possible choices of balancing the relative importance of energy, force, and stress errors, as well as computational costs and data set diversity. Our benchmarking shows that the choice of acquisition function strongly controls the evolution of errors in the iterative training process. We believe that the performance of the best acquisition function, `weighted_average`, is partially due to factoring in the computational cost of obtaining the DFT data. In general, from a purely computational perspective, it is important to incorporate the DFT cost in the AL strategy.

With the aim of reducing computational costs, we benchmarked batched AL and found that batches of structures could be selected and added to the model in a retraining step without significant impact upon the AL algorithm trajectory. k-means clustering can be done on the energy descriptor input space, which separates structures largely based on their atomic density.

While the effect of clustering on batches up to $L = 10$ is non-negligible, it is much less significant than the choice of acquisition function.

The connection of force uncertainties or errors to those of properties of interest in molecular dynamics simulations is nontrivial but vital. It is a matter of future work to propagate the Bayesian uncertainties to macroscale properties and inform the active learning algorithms by acquisition functions targeted on reducing uncertainty for those macroscale properties.

Scripts for running the AL algorithms in this study are included as an example in the FitSNAP GitHub repository (`github.com/FitSNAP/FitSNAP`) under `examples/library/bayesian_active_learning`. The data sets used in this paper are those of the Ta, InP, and WBe data sets also in the `examples/` folder. The active learning script folder contains the Ta dataset already broken into the test/train split used in this paper.

Acknowledgement. We acknowledge funding support from the U.S. Department of Energy, Office of Fusion Energy Sciences (OFES) under Field Work Proposal Number 20-023149. This article has been authored by employees of National Technology & Engineering Solutions of Sandia, LLC under Contract No. DE-NA0003525 with the U.S. Department of Energy (DOE). The employees co-own right, title and interest in and to the article and are responsible for its contents. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this article or allow others to do so, for United States Government purposes. The DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan `https://www.energy.gov/downloads/doe-public-access-plan`.

References

[1] Yunxing Zuo, Chi Chen, Xiangguo Li, Zhi Deng, Yiming Chen, Jörg Behler, Gábor Csányi, Alexander V. Shapeev, Aidan P. Thompson, Mitchell A. Wood, and Shyue Ping Ong. Performance and cost assessment of machine learning interatomic potentials. *The Journal of Physical Chemistry A*, 124(4):731–745, 2020. PMID: 31916773.

[2] Albert P. Bartók, Mike C. Payne, Risi Kondor, and Gábor Csányi. Gaus-

sian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.*, 104:136403, Apr 2010.

[3] Roman M. Balabin and Ekaterina I. Lomakina. Support vector machine regression (LS-SVM)—an alternative to artificial neural networks (ANNs) for the analysis of quantum chemistry data? *Phys. Chem. Chem. Phys.*, 13:11710–11718, 2011.

[4] Jörg Behler. Representing potential energy surfaces by high-dimensional neural network potentials. *Journal of Physics: Condensed Matter*, 26(18):183001, April 2014.

[5] Alexander V. Shapeev. Moment tensor potentials: A class of systematically improvable interatomic potentials. *Multiscale Modeling & Simulation*, 14(3):1153–1173, 2016.

[6] Z. Xie and J. M. Bowman. Permutationally invariant polynomial basis for molecular energy surface fitting via monomial symmetrization. *Journal of Chemical Theory and Computation*, 6(1):26–34, 2010.

[7] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O. Anatole von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.*, 108:058301, Jan 2012.

[8] Aidan P Thompson, Laura P Swiler, Christian R Trott, Stephen M Foiles, and Garritt J Tucker. Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials. *Journal of Computational Physics*, 285:316–330, 2015.

[9] David Montes de Oca Zapiain, Mitchell A. Wood, Nicholas Lubbers, Carlos Z. Pereyra, Aidan P. Thompson, and Danny Perez. Training data selection for accuracy and transferability of interatomic potentials. *NPJ Computational Materials*, 8(1), 2022.

[10] Burr Settles. Active learning literature survey. Technical report, 2010.

[11] Giulio Imbalzano, Andrea Anelli, Daniele Giofré, Sinja Klees, Jörg Behler, and Michele Ceriotti. Automatic selection of atomic fingerprints and reference configurations for machine-learning potentials. *The Journal of Chemical Physics*, 148(24):241730, 04 2018.

[12] Ganesh Sivaraman, Anand Narayanan Krishnamoorthy, Matthias Baur, Christian Holm, Marius Stan, Gábor Csányi, Chris Benmore, and Alvaro Vazquez-Mayagoitia. Machine-learned interatomic potentials by active learning: amorphous and liquid hafnium dioxide. *npj Computational Materials*, 6(104), 2020.

[13] Noam Bernstein, Gábor Csányi, and Volker L. Deringer. De novo exploration and self-guided learning of potential-energy surfaces. *npj Computational Materials*, 5(99), 2019.

[14] 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), 2021.

[15] Evgeny V. Podryabinkin and Alexander V. Shapeev. Active learning of linearly parametrized interatomic potentials. *Computational Materials Science*, 140:171 – 180, 2017.

[16] Ivan S. Novikov, Yury V. Suleimanov, and Alexander V. Shapeev. Automated calculation of thermal rate coefficients using ring polymer molecular dynamics and machine-learning interatomic potentials with active learning. *Phys. Chem. Chem. Phys.*, 20:29503–29512, 2018.

[17] Ivan S. Novikov, Alexander V. Shapeev, and Yury V. Suleimanov. Ring polymer molecular dynamics and active learning of moment tensor potential for gas-phase barrierless reactions: Application to S + H2. *The Journal of Chemical Physics*, 151(22):224105, 2019.

[18] Evgeny V. Podryabinkin, Evgeny V. Tikhonov, Alexander V. Shapeev, and Artem R. Oganov. Accelerating crystal structure prediction by machine-learning interatomic potentials with active learning. *Phys. Rev. B*, 99:064114, Feb 2019.

[19] Ivan S. Novikov, Konstantin Gubaev, Evgeny Podryabinkin, and Alexander V. Shapeev. The MLIP package: moment tensor potentials with MPI and active learning. *Machine Learning: Science and Technology*, 2:025002, 2021.

[20] Konstantin Gubaev, Evgeny V. Podryabinkin, and Alexander V. Shapeev. Machine learning of molecular properties: Locality and active learning. *The Journal of Chemical Physics*, 148(24):241727, 2018.

[21] Max Hodapp and Alexander V. Shapeev. In operando active learning of interatomic interaction during large-scale simulations. *Machine Learning: Science and Technology*, 1:045005, 2020.

[22] Gabor Csányi, T. Albaret, M. C. Payne, and A. De Vita. "Learn on the Fly": A Hybrid Classical and Quantum-Mechanical Molecular Dynamics Simulation. *Phys. Rev. Lett.*, 93:175503, Oct 2004.

[23] Jonathan Vandermause, Steven B. Torrisi, Simon Batzner, Yu Xie, Lixin Sun, Alexie M. Kolpak, and Boris Kozinsky. On-the-fly active learning of interpretable Bayesian force fields for atomistic rare events. *npj Computational Materials*, 6(20), 2020.

[24] Ryosuke Jinnouchi, Kazutoshi Miwa, Ferenc Karsai, Georg Kresse, and Ryoji Asahi. On-the-fly active learning of interatomic potentials for large-scale atomistic simulations. *The Journal of Physical Chemistry Letters*, 11(17):6946–6955, 2020.

[25] Yu Xie, Jonathan Vandermause, Lixin Sun, Andrea Cepellotti, and Boris Kozinsky. Bayesian force fields from active learning for simulation of inter-dimensional transformation of stanene. *npj Computational Materials*, 7(40), 2021.

[26] Anders Johansson, Yu Xie, Cameron J. Owen, Jin Soo Lim, Lixin Sun, Jonathan Vandermause, and Boris Kozinsky. Micron-scale heterogeneous catalysis with Bayesian force fields from first principles and active learning. 2022.

[27] Yu Xie, Jonathan Vandermause, Senja Ramakers, Nakib Protik, Anders Johansson, and Boris Kozinsky. Uncertainty-aware molecular dynamics from Bayesian active learning for phase transformations and thermal transport in SiC. *npj Computational Materials*, 9(36), 2023.

[28] Linfeng Zhang, De-Ye Lin, Han Wang, Roberto Car, and Weinan E. Active learning of uniformly accurate interatomic potentials for materials simulation. *Phys. Rev. Materials*, 3:023804, Feb 2019.

[29] Wonseok Jeong, Dongsun Yoo, Kyuhyun Lee, Jisu Jung, and Seungwu Han. Atomic-resolution uncertainty estimation for neural network potentials using a replica ensemble. *J Phys Chem Lett.*, 11(15):6090–6096, 2020.

[30] Christoph Schran, Krystof Brezina, and Ondrej Marsalek. Committee neural network potentials control generalization errors and enable active learning. *The Journal of Chemical Physics*, 153(10):104105, 2020.

[31] Justin S. Smith, Ben Nebgen, Nicholas Lubbers, Olexandr Isayev, and Adrian E. Roitberg. Less is more: sampling chemical space with active learning. *The Journal of Chemical Physics*, 148(24):241733, 2018.

[32] Justin S. Smith, Benjamin Nebgen, Nithin Mathew, Jie Chen, Nicholas Lubbers, Leonid Burakovsky, Sergei Tretiak, Hai Ah Nam, Timothy Germann, Saryu Fensin, and Kipton Barros. Automated discovery of a robust interatomic potential for aluminum. *Nature Communications*, 12(1257), 2021.

[33] Giulio Imbalzano, Yongbin Zhuang, Venkat Kapil, Kevin Rossi, Edgar A. Engel, Federico Grasselli, and Michele Ceriotti. Uncertainty estimation for molecular dynamics and sampling. *The Journal of Chemical Physics*, 154(7):074102, 2021.

[34] Nathan Wilson, Daniel Willhelm, Xiaoning Qian, Raymundo Arróyave, and Xiaofeng Qian. Batch active learning for accelerating the development of interatomic potentials. *Computational Materials Science*, 208:111330, 2022.

[35] Søren L. Frederiksen, Karsten W. Jacobsen, Kevin S. Brown, and James P. Sethna. Bayesian ensemble approach to error estimation of interatomic potentials. *Physical Review Letters*, 93(16):1–4, 2004.

[36] John C. Snyder, Matthias Rupp, Katja Hansen, Klaus-Robert Müller, and Kieron Burke. Finding density functionals with machine learning. *Phys. Rev. Lett.*, 108:253002, Jun 2012.

[37] Joshua J. Gabriel, Noah H. Paulson, Thien C. Duong, Francesca Tavazza, Chandler A. Becker, Santanu Chaudhuri, and Marius Stan. Uncertainty Quantification in Atomistic Modeling of Metals and Its Effect on Mesoscale and Continuum Modeling: A Review. *JOM (The Journal of The Minerals, Metals & Materials Society)*, 73:149–163, 2021.

[38] Leonid Kahle and Federico Zipoli. Quality of uncertainty estimates from neural network potential ensembles. *Phys. Rev. E*, 105:015311, Jan 2022.

[39] Marco Fronzi, Olexandr Isayev, David A. Winkler, Joseph G. Shapter, Amanda V. Ellis, Peter C. Sherrell, Nick A. Shepelin, Alexander Corletto, and Michael J. Ford. Active Learning in Bayesian Neural Networks for Bandgap Predictions of Novel Van der Waals Heterostructures. *Advanced Intelligent Systems*, 3(11):2100080, 2021.

[40] Albert Zhu, Simon Batzner, Albert Musaelian, and Boris Kozinsky. Fast Uncertainty Estimates in Deep Learning Interatomic Potentials. *The Journal of Chemical Physics*, 158(16):164111, 04 2023.

[41] Mitchell A Wood and Aidan P Thompson. Extending the accuracy of the SNAP interatomic potential form. *The Journal of chemical physics*, 148(24):241721, 2018.

[42] Dmitrii A. Varshalovich, Anatolij N. Moskalev, and Valerii K. Khersonskii. *Quantum theory of angular momentum*. World Scientific, 1988.

[43] D. S. Sivia and J. Skilling. *Data Analysis: A Bayesian Tutorial, Second Edition*. Oxford University Press, 2006.

[44] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[45] A. Rohskopf, C. Sievers, N. Lubbers, M.A. Cusentino, J. Goff, J. Janssen, M. McCarthy, D. Montes Oca de Zapiain, S. Nikolov, K. Sargsyan, D. Sema, E. Sikorski, L. Williams, A.P. Thompson, and M.A. Wood. FitSNAP: Atomistic machine learning with LAMMPS. *Journal of Open Source Software*, 8(84):5118, 2023.

[46] M. A. Cusentino, M. A. Wood, and A. P. Thompson. Explicit multielement extension of the spectral neighbor analysis potential for chemically complex systems. *The Journal of Physical Chemistry A*, 124(26):5456–5464, 2020. PMID: 32432859.

[47] M. A. Wood, M. A. Cusentino, B. D. Wirth, and A. P. Thompson. Data-driven material models for atomistic simulation. *Phys. Rev. B*, 99:184305, May 2019.