Computational Materials Science 232 (2024) 112655

Contents lists available at ScienceDirect

Computational Materials Science

journal homepage: www.elsevier.com/locate/commatsci

Full length article

Spline-based neural network interatomic potentials: Blending classical and machine learning models

Joshua A. Vita, Dallas R. Trinkle *

Department of Materials Science and Engineering, University of Illinois Urbana-Champaign, Urbana, IL 61801, USA

ARTICLE INFO

Keywords:
Interatomic potential
Splines
Machine learning
Interpretability

ABSTRACT

While machine learning (ML) interatomic potentials (IPs) are able to achieve accuracies nearing the level of noise inherent in the first-principles data to which they are trained, it remains to be shown if their increased complexities are strictly necessary for constructing high-quality IPs. In this work, we introduce a new MLIP framework which blends the simplicity of spline-based MEAM (s-MEAM) potentials with the flexibility of a neural network (NN) architecture. The proposed framework, which we call the spline-based neural network potential (s-NNP), is a simplified version of the traditional NNP that can be used to describe complex datasets in a computationally efficient manner. We demonstrate how this framework can be used to probe the boundary between classical and ML IPs, highlighting the benefits of key architectural changes. Furthermore, we show that using spline filters for encoding atomic environments results in a readily interpreted embedding layer which can be coupled with modifications to the NN to incorporate expected physical behaviors and improve overall interpretability. Finally, we test the flexibility of the spline filters, observing that they can be shared across multiple chemical systems in order to provide a convenient reference point from which to begin performing cross-system analyses.

1. Introduction and background

In the fields of computational materials science and chemistry, machine learning interatomic potentials (MLIPs) are rapidly becoming an essential tool for running high-fidelity atomic-scale simulations. Notably, major breakthroughs in the field have typically been marked by the development of new methods for encoding the atomic environments into machine-readable descriptors, or using architectures from other fields of machine learning to improve regression from descriptors into energies and atomic forces. The seminal work using atom-centered symmetry functions (ACSFs) [1] as the encoder, followed by "smooth overlap of atomic position" (SOAP) descriptors [2], and, most recently, equivariant message-passing networks [3] being milestones of particular importance in the field. In parallel with these improvements to the encoding functions of MLIPs has been the development of new regression tools, favored either for the simplicity gained through the use of linear combinations of basis functions (e.g., [4–6]) or the accuracy gained by increasing the effective cutoff radius of the model using message-passing neural networks (e.g., [7–9]). While countless other models have been proposed using combinations or variations of different methods (an incomplete list: [10–18]), the key insights remain the same: interatomic potentials can be greatly improved by leveraging architectures and optimization strategies drawn from other machine learning and deep learning applications.

Despite the success of these MLIPs, there has been a persistent need in the community for the continued development of low-cost classical IPs, particularly for use with large-scale simulations [19–23]. Some of the major benefits that classical IPs [24–31] have over MLIPs are their low computational costs, strong foundations in known physics, and a history of scientific research analyzing their behaviors and theoretical limitations. Although improvements are being made to the speeds of MLIPs, the stark differences between classical and ML IPs in terms of size, design, and overall complexity make it difficult to leverage the well-established tools and knowledge from classical models in order to further improve their ML counterparts.

In this work, we develop a new IP model whose hyper-parameters can be tuned to transition smoothly between low-cost, low-complexity classical models and full MLIPs. By basing the proposed model off of a classical spline-based MEAM potential, then extending it using a basic ML architecture, we enable direct comparisons to well-established classical forms as well as modern ML models. These results help to bridge the gap between the two classes of models and highlight methods for improving speeds, interpretability, and transferability of MLIPs.

* Corresponding author.
E-mail addresses: jvita2@illinois.edu (J.A. Vita), dtrinkle@illinois.edu (D.R. Trinkle).

https://doi.org/10.1016/j.commatsci.2023.112655
Received 21 September 2023; Received in revised form 10 November 2023; Accepted 10 November 2023
Available online 17 November 2023

1.1. s-MEAM

The model developed in this paper builds heavily upon the spline-based MEAM potential [32] ("s-MEAM"), which we describe here in order to provide sufficient background to understand the new model architecture proposed in this work. The s-MEAM potential is a spline-based version of the popular analytical MEAM potential [31] that was intended to provide additional flexibility to the model while maintaining the same overall functional form. In the s-MEAM formalism, the energy E_i of a given atom i is written as:

$$\begin{aligned} E &= \sum_i E_i = \sum_i \left[U_{c_i}(n_i) + \sum_{j<i} \phi_{c_j}(r_{ij}) \right] \\ n_i &= \sum_{j\neq i} \rho_{c_j}(r_{ij}) \\ &\quad + \sum_{\substack{j<k,\\ j,k\neq i}} f_{c_j}(r_{ij}) f_{c_k}(r_{ik}) g_{c_j c_k}\left[\cos(\theta_{jik})\right]. \end{aligned} \tag{1}$$

In Eq. (1) the energy of the ith atom, E_i, is composed of a pair term (ϕ) and an embedding energy contribution (U) for a given "electron density" n_i, where all five functions (ϕ, U, ρ, f, and g) are represented using cubic Hermite splines. The pair term is calculated by summing over pair distances $r_{ij} = |\vec{r}_j - \vec{r}_i|$ between each atom i and its neighbors j (with r_{ij} less than a chosen cutoff distance, r_c). The electron density n_i is further decomposed into 2-body (ρ) and 3-body (products of f and g) contributions. The 2-body term is similar to the summation over ϕ, while the 3-body term is computed by summing over the product of three spline functions that take r_{ij}, r_{ik}, and $cos(\theta_{jik})$ as inputs, where θ_{jik} is the bond angle formed by atoms i, j, and k with i at the center. The subscripts on the functions indicate that separate splines are used for evaluation depending on the chemistries c_i, c_j, and c_k of atoms i, j, and k (e.g., g_{AA} for A–A bonds, g_{AB} for A–B bonds, etc.).

In order to facilitate comparisons between s-MEAM and the model that will be proposed in this work, we will first define two new functions

$$\begin{aligned} G^{\alpha}_{3,i}(r_{ij}, r_{ik}, \cos\theta_{jik}) &= \\ &\sum_{\substack{j<k\\ j,k\neq i}} f^{\alpha}_{c_j}(r_{ij}) f^{\alpha}_{c_k}(r_{ik}) g^{\alpha}_{c_{jk}}\left(\cos(\theta_{jik})\right) \\ G^{\beta}_{2,i}(r_{ij}) &= \sum_{j\neq i} \rho^{\beta}_{c_j}(r_{ij}), \end{aligned} \tag{2}$$

then re-write Eq. (1) as

$$\begin{aligned} E &= \sum_i \left[U_{c_i}(n_i) + \frac{1}{2} G^0_{2,i} \right] \\ n_i &= \sum_{\beta=1}^{N_2} G^{\beta}_{2,i} + \sum_{\alpha=1}^{N_3} G^{\alpha}_{3,i} \end{aligned}, \tag{3}$$

where $N_2 = 1$ and $N_3 = 1$. We will henceforth refer to $G^{\alpha}_{3,i}$ and $G^{\beta}_{2,i}$ as 3-body and 2-body "spline filters" respectively, in acknowledgment of the fact that they can be thought of as filters that characterize the local environment around atom i in order to produce a scalar atomic environment descriptor n_i. In Eq. (3) we have introduced summations over the superscripts α and β which currently only take on a single value of 1, but will be used later to denote different filters.

1.2. NNP

Shown to be universal function approximators [33], neural networks (NNs) are provably more flexible than classical IPs which use explicit analytical forms. Because of this, a sufficiently large NN would be expected to be able to accurately reproduce an arbitrary potential energy surface, assuming that it properly accounted for long-range interactions, was provided with enough fitting data, and did not suffer from limitations due to trainability. The original Behler–Parrinello NNP [1] was one of the first successful applications of NNs towards practical systems, where the atomic energy of atom i is written as

$$\begin{aligned} E &= \sum_i N_{c_i}(\vec{D}_i) \\ \vec{D}_i &= \langle D^1_{3,i}, \ldots, D^{N_3}_{3,i}, D^1_{2,i}, \ldots, D^{N_2}_{2,i} \rangle. \end{aligned} \tag{4}$$

In Eq. (4), N_{c_i} is a neural network, c_i is the element type of atom i, and $\vec{D}_i$ is the atom-centered symmetry function (ACSF) descriptor [1] of atom i. The ACSF local environment descriptor is comprised of radial symmetry functions

$$D^{\beta}_{2,i}(r_{ij}) = \sum_{j\neq i} e^{-\eta^{\beta}(r_{ij} - R^{\beta}_s)^2} v_c(r_{ij}), \tag{5}$$

parameterized by η^{β} for changing the width of the Gaussian distribution, and R^{β}_s to shift the distribution. A smooth cutoff function $v_c(r_{ij})$ is used with the form:

$$v_c(r_{ij}) = \begin{cases} 0.5 \times \left[\cos\frac{\pi r_{ij}}{r_c} + 1\right], & \text{if } r_{ij} \leq r_c \\ 0, & \text{if } r_{ij} > r_c. \end{cases} \tag{6}$$

Angular contributions are accounted for using the angular symmetry functions

$$\begin{aligned} D^{\alpha}_{3,i}(r_{ij}, r_{ik}, r_{jk}, \theta_{jik}) &= \\ &2^{1-\zeta^{\alpha}} \sum_{j,k\neq i} (1 + \lambda^{\alpha}\cos\theta_{jik})^{\zeta^{\alpha}} \\ &\times e^{-\eta^{\alpha}(r^2_{ij} + r^2_{ik} + r^2_{jk})} v_c(r_{ij}) v_c(r_{ik}) v_c(r_{jk}). \end{aligned} \tag{7}$$

Multiple radial and angular symmetry functions are constructed by making N_2 choices for η^{β} and R^{β}_s, and N_3 choices for ζ^{α}, λ^{α} ($= \pm 1$), and η^{α}. The evaluations of all of these symmetry functions are then concatenated together into a single vector $\vec{D}_i$ which is passed through a feed-forward neural network.

Obvious parallels can be drawn between the NNP form in Eq. (4) and the re-written form of s-MEAM shown in Eq. (3). What differentiates NNP from s-MEAM, however, are the details regarding the construction of the local descriptors, and the form of the embedding function. where s-MEAM uses trainable spline filters for both the descriptor and the embedding function, NNP uses ACSFs and an NN respectively. Although an NN would have an increased fitting capacity over the U_{c_i} splines used in Eq. (3), there are many similarities between the ACSF descriptors and the spline filters described in Eq. (2). For example, the 2-body components of an ACSF descriptor, which are constructed by evaluating $D^{\beta}_{2,i}$ with multiple radial shifts R^{β}_s for each neighboring atom j, can be viewed as basis functions used for interpolating the desired range of atomic distances. This is conceptually related to how the basis functions of cubic Hermite splines allow $G^{\beta}_{2,i}$ to interpolate over its domain as well. A similar argument can be made relating the angular components of ACSFs to the 3-body filters $G^{\alpha}_{3,i}$, where Eqs. (2) and (5) both multiply functions of pair distances ($f^{\alpha}_{c_k}(r_{ij})$ and $f^{\alpha}_{c_k}(r_{ik})$) by a function of the triplet angle ($g^{\alpha}_{c_{jk}}(\cos(\theta_{jik}))$). The ANI model [34], which will be used in this work for comparison to the model which we developed, is nearly identical to the NNP form described above, with the modifications that only a single η^{β} is used, and Eq. (7) is altered to introduce both radial (R^{α}_s) and angular (θ^{α}_s) shifts:

$$\begin{aligned} D^{\alpha}_{3,i}(r_{ij}, r_{ik}, \theta_{jik}) &= \\ &2^{1-\zeta^{\alpha}} \sum_{j,k\neq i} [1 + \cos(\theta_{jik} - \theta^{\alpha}_s)]^{\zeta^{\alpha}} \\ &\times e^{-\eta^{\alpha}\left(\frac{r_{ij} + r_{ik}}{2} - R^{\alpha}_s\right)} v_c(r_{ij}) v_c(r_{ik}). \end{aligned} \tag{8}$$

2. Methods

2.1. s-NNP

The main contribution of this work is the development of a spline-based neural network potential, outlined in Fig. 1, which we call the

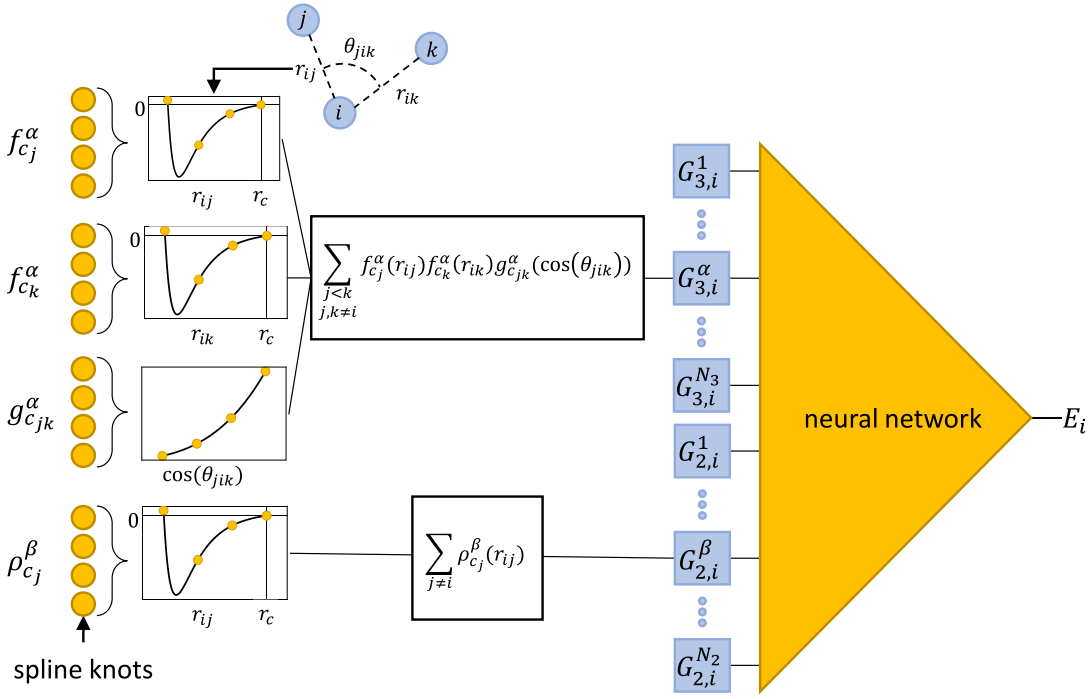

Fig. 1. A diagram of the s-NNP model architecture. The spline knots parameterize individual 2-body ($f^{\alpha}_{c_j}$, $f^{\alpha}_{c_k}$, and $\rho^{\beta}_{c_j}$) or 3-body ($g^{\alpha}_{c_{jk}}$) splines, which take as input the pair distances r_{ij}, r_{ik} or triplet angles θ_{jik} for all neighboring atoms j and k within a cutoff around each atom i. The outputs of the splines are then summed (after multiplying $f^{\alpha}_{c_j}$, $f^{\alpha}_{c_k}$, and $g^{\alpha}_{c_{jk}}$ together as described in Eq. (2)) over all pairs or triplets. N_3 3-body and N_2 2-body filters are used (each parameterized by their own sets of knots, and indexed based upon the chemistries c_j, c_k, and c_{jk} of each interaction), and the outputs are concatenated into a single vector which is propagated through a feed-forward neural network. All radial splines are pinned to have a value of zero at their cutoff distance. Unless otherwise specified, the s-NNP uses a CELU activation layer after every linear layer in the network except for the output layer.

"s-NNP" (short for "spline-NNP"). The s-NNP framework is intended to extend the fitting capacity of the s-MEAM class of potentials while maintaining the high interpretability and speed provided by the use of splines. s-NNP can be thought of as an s-MEAM potential with two critical changes: first, N_3 and N_2 in Eq. (3), the numbers of $G^{\alpha}_{3,i}$ and $G^{\beta}_{2,i}$ spline filters, are taken to be hyper-parameters of the model; and second, the "embedding" spline U_i in an s-MEAM model is replaced by a fully-connected neural network. By including additional spline filters, the s-NNP is able to describe the local environment around an atom with increasingly fine resolution (since each spline can be thought of as a basis function for interpolating the atomic energies). The introduction of a neural network allows the model to achieve much more complex mappings into atomic energies than would be possible with the cubic splines U_{c_i}. In the s-NNP formalism, the energy E_i of a given atom i is then written as

$$\begin{aligned} E &= \sum_i N(\vec{G}_i), \\ \vec{G}_i &= \langle G^1_{3,i}, \ldots, G^{N_3}_{3,i}, G^1_{2,i}, \ldots, G^{N_2}_{2,i} \rangle. \end{aligned} \tag{9}$$

where N is a neural network, and $\vec{G}_i$ is a vector of length $(N_3 + N_2)$. Notice that each component of $\vec{G}_i$ is computed by evaluating a different 3- or 2-body spline filter for the local environment of atom i. The parameters of an s-NNP that are trained during fitting are the positions of the knots of the f^{α}, g^{α}, and ρ^{β} splines from Eq. (2), as well as the weights and biases in the neural network N. The hyper-parameters of the model are N_3, N_2, the number of knots in each spline, the number of layers in N, and the number of hidden nodes in each of those layers.

One benefit of the s-NNP framework is that it is closely related to both classical and machine learning interatomic potentials (see Section 4.2 for more discussion), making it possible to easily probe the performance gap between the two. For example, many classical potentials (LJ [24], EAM [25], MEAM [31], Stillinger–Weber [35], Tersoff [27], and Buckingham [26]) could be reformulated as s-NNP potentials with very few filters ($N_3 \in [0, 1]$, $N_2 \in [1, 2]$) and custom embedding functions instead of a neural network (though given the universal approximation theorem, these embedding functions could be represented using an NN as well). Because of this, we can easily construct spline-based "classical" models by adjusting N_3 and N_2, but not including a neural network, then compare them directly to MLIPs by subsequently attaching networks with varying depths and widths. See Section 3 in the Results for details of such a study.

2.2. Interpretability improvements

A central tenet of constructing interpretable models is designing the model architecture in a way that helps isolate the contributions of specific parameter sets to the final model predictions. With this in mind, we therefore propose the five modifications described in Fig. 2. Although the modifications proposed here are only discussed in the context of s-NNP, they can also be applied to many other existing MLIP frameworks.

When applying all of the modifications described in Fig. 2, the full form of s-NNP is written as:

$$\begin{aligned} E &= \sum_i E_i = \sum_i \left[E_{\text{lin},i} + E_{\text{net},i} \right] \\ E_{\text{lin},i} &= E_{\text{sc},i} + G^{\text{short}}_{2,i} + b \\ E_{\text{sc},i} &= \sum_{n=1}^{N_3} G^n_{3,i} + \sum_{n=1}^{N_2} G^n_{2,i} \\ E_{\text{net},i} &= \sigma\left[N_{\text{no-bias}}(\vec{G}_i) \right], \end{aligned} \tag{10}$$

where $E_{\text{sc},i}$ as a "skip connection" term (as used in other DL fields [36]), $G^{\text{short}}_{2,i}$ is a short-range 2-body filter, b is a trainable isolated atom energy, σ is the Softplus activation function $\sigma(x) = \log(1 + e^x)$, and $N_{\text{no-bias}}$ is a neural network with no bias terms on any of its layers. Note that $E_{\text{lin},i}$ encompasses all of the terms which are linearly dependent

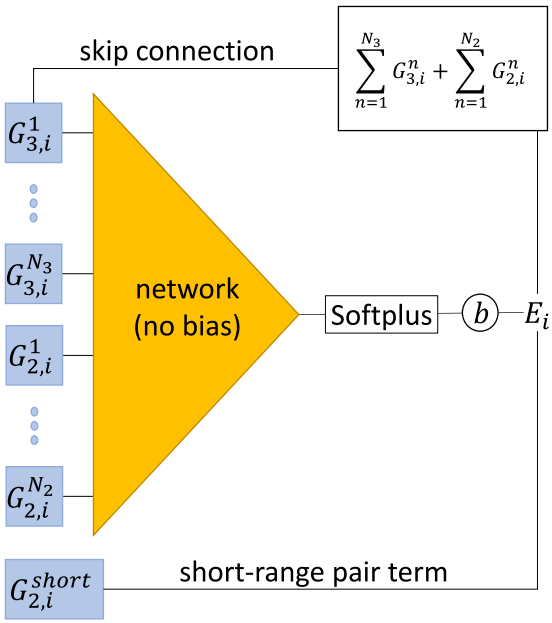

Fig. 2. Modifications to the s-NNP form described in Fig. 1 for improving interpretability. The four modifications are: (1) adding skip connections; (2) removing the internal network bias, and adding an external bias, b; (3) wrapping the network outputs in a Softplus activation function; and (4) introducing a single short-range pair term using a 2-body filter. The short-range pair term is only allowed to be non-zero up to a cutoff of 2.5 Å.

upon the spline filters, and $E_{\text{net},i}$ captures the non-linear dependence. While a more in-depth discussion of the effects of each of these modifications can be found in Section 4.1, the practical result of Eq. (10) is that the spline filter visualizations like those shown in Fig. 3 can be intuitively understood. For example, negative regions of the filters will usually correspond to negative E_i, and positive regions of the filters will always correspond to positive E_i. While Eq. (10) still suffers from some drawbacks, predominantly associated with the non-linear effects of $N_{\text{no-bias}}(\vec{G}_i)$, we believe that it strikes a good balance between the accuracy achieved through a NN architecture, and the interpretability characteristic of classical potentials.

2.3. Benchmarking dataset: Al

In order to test our framework, in Section 3.1 we fit s-NNP models to the aluminum dataset from Smith et al. [37], which serves as a good benchmarking dataset due to its size and configurational complexity. This dataset was built using an active learning technique for generating interatomic potential training data [38]. It has also been shown to contain extremely diverse atomic environments, such that the ANI model [39] which was originally trained to the dataset was tested for use in shock simulations. After removing duplicate calculations (identical atomic positions and computed energies/forces, generated at different active learning steps), and one outlier configuration with particularly high forces, the final dataset consisted of 5751 unique configurations (744,356 atoms total) with their corresponding DFT-computed energies and forces. A 90:10 train-test split was performed to partition the dataset into training/testing data. This data can be obtained from the original source [40].

2.4. Additional datasets: Cu, Ge, Mo

As a test of the flexibility of the spline filters, in Section 3.3 we trained an s-NNP model simultaneously to the Al dataset described in Section 2.3 and the Cu, Ge, and Mo datasets from Zuo et al. [41]. Each dataset from [41] was manually constructed and contains the ground state structure for the given element, strained supercells, slab structures, *ab initio* molecular dynamics (AIMD) sampling of supercells at different temperatures (300 K and 0.5×, 0.9×, 1.5×, and 2.0× the melting point), and AIMD sampling of supercells with single vacancies at 300 K and 2.0× the melting point. On average, each training dataset includes approximately 250 structures for a total of 27,416 atoms (Cu), 14,072 atoms (Ge), and 10,087 atoms (Mo). A detailed summary of the contents of the datasets can be found in [41]. This data can be obtained from the original source [42].

In order to attempt to balance the combined dataset, we took a random subset comprised of 20% of the Al dataset in addition to the full Cu, Ge, and Mo datasets, resulting in a total training set size of 1271 Al configurations, 262 Cu, 228 Ge, and 194 Mo. Another logical choice for constructing the multi-element training set would be to further downsample the Al dataset to better balance the relative concentrations of each element type. However, we observed that doing so resulted in worse property predictions for Al, presumably due to the large number of high-energy configurations in the original Al dataset making random sampling of low-energy configurations relevant to the properties of interest unlikely. In all cases, the atomic energies of each element type were shifted by the average energy of that type before training.

3. Results

3.1. Benchmarking tests

Using the s-NNP framework, we fit a collection of models to the Al dataset to probe the effects of increasing model capacity in two distinct ways: first by increasing the number of spline filters, and second by increasing the size of the network used for mapping filter outputs to energies. Table 1 outlines the architectures and accuracies of the trained models, grouped by key architectural changes and sorted by total number of parameters. The trained s-NNP models can be conceptually broken down into three main groups, each highlighting the effects of specific architectural changes: (1) "(N_2, N_3) linear" models, increasing the number of splines when no neural network is used; (2) "(N_2, N_3) l =*" models, increasing the number of splines and network size; and (3) introducing the interpretability improvements described in Section 2.2.

The results in Table 1 show multiple avenues for systematically improving the performance of an s-NNP model, though each method appears to experience a saturation point beyond which the model suffers from diminishing returns as complexity increases. Using the "$(1, 1)$ linear" model as a baseline, we see that increasing N_3 from 1 to 8 can monotonically decrease both the energy and the force errors. Increasing N_2 has no significant effect on the accuracy of the linear models, which is consistent with the notion that a linear combination of cubic Hermite splines can be represented using a single spline.

The introduction of a neural network enables the model to better utilize the additional spline filters, leading to significant improvement of the "$(1, 2)$ $l = 3$" model over any of the linear models. Increasing the number of spline filters, and subsequently increasing the network width and depth to maintain a "funnel-like" structure (decreasing layer width with increasing depth) can also lead to steady improvements. However, with increasing model size we began to see a commensurate increase in training difficulty, often resulting in larger models with higher errors than what might be expected based on the performance of their smaller counterparts (e.g., "(2,4) $l = 6$" compared to "(2,4) $l = 5$", or "(1,8) $l = 5$, int. wide" compared to "(1,8) $l = 5$, int. all"). Notably, the interpretability improvements from Section 2.2 did not hurt model performance, demonstrating that accuracy and interpretability are not strictly opposing attributes of a model. Based on the results shown in Table 1, we will use the "(1,8) $l = 5$, int. all" for all experiments and analyses in the remainder of this paper.

Table 1
Fitting results comparing linear s-NNP models, "(N_2, N_3) linear", s-NNP models with neural networks, "(N_2, N_3) l =*", s-NNP with interpretability improvements "(1,8) l = 5, int. *", and the ANI model from [37], "ANI". The (N_2, N_3) notation for the spline layers indicates that N_2 2-body and N_3 3-body spline filters were used. Network architectures are denoted as a tuple of integers ("Network" column) specifying the number of nodes in each hidden layer of the model. Although the training/testing errors decrease significantly when adding additional splines to the linear models, their performance appears to begin to saturate with the "(1,8) linear" model. Model performance only begins to be competitive with the ANI results upon the inclusion of a neural network with sufficient depth, which allows for non-linear combinations of spline outputs. While the "(1,8) l = 5, int. skip" model only adds the use of skip connections, the "(1,8) l = 5, int. all" and "(1,8) l = 5, int. wide" use all of the interpretability improvements discussed in Sections 2.2 and 4.1. Note that this means that the single 2-body filter specified using the "(1,8)" notation is the short-range pair term described in Fig. 2, and that there are therefore no 2-body filters being passed through the network. Testing errors for the "ANI" model were taken directly from [37], which did not report training errors. Note that the ANI model uses ensemble-averaging over 8 networks, which they report yields energy and force errors that are "20% and 40% smaller, respectively, compared to a single ANI model".

Model name	(N_2, N_3)	Network	Parameters (splines, network)	E_{RMSE} (meV/atom)		F_{RMSE} (eV/Å)	
				Train	Test	Train	Test
(1, 1) linear	(1, 1)	–	(66, 0)	74	78	0.76	0.82
(1, 2) linear	(1, 2)	–	(110, 0)	56	56	0.54	0.58
(1, 4) linear	(1, 4)	–	(198, 0)	40	41	0.46	0.49
(2, 4) linear	(2, 4)	–	(220, 0)	38	36	0.46	0.50
(1, 8) linear	(1, 8)	–	(374, 0)	32	34	0.42	0.46
(1, 2) l = 3	(1, 2)	(3, 2, 1)	(110, 23)	44	44	0.33	0.35
(1, 4) l = 5	(1, 4)	(5, 4, 3, 2, 1)	(198, 80)	10	11	0.22	0.23
(2, 4) l = 4	(2, 4)	(6, 3, 2, 1)	(220, 74)	22	21	0.21	0.21
(2, 4) l = 5	(2, 4)	(6, 4, 3, 2, 1)	(220, 96)	8.7	9.0	0.19	0.20
(2, 4) l = 6	(2, 4)	(6, 5, 4, 3, 2, 1)	(220, 127)	10	12	0.24	0.25
(1, 8) l = 5	(1, 8)	(9, 8, 4, 2, 1)	(374, 219)	7.5	6.5	0.13	0.13
(1, 8) l = 5, int. skip	(1, 8)	(9, 8, 4, 2, 1)	(374, 219)	5.5	4.6	0.15	0.15
(1, 8) l = 5, int. all	(1, 8)	(9, 8, 4, 2, 1)	(374, 219)	5.5	5.9	0.12	0.12
(1, 8) l = 5, int. wide	(1, 8)	(128, 64, 32, 16, 1)	(374, 11,793)	5.9	5.1	0.13	0.14
ANI [37]	–	(96, 96, 64, 1)	(–, 15,585)	–	1.9	–	0.06

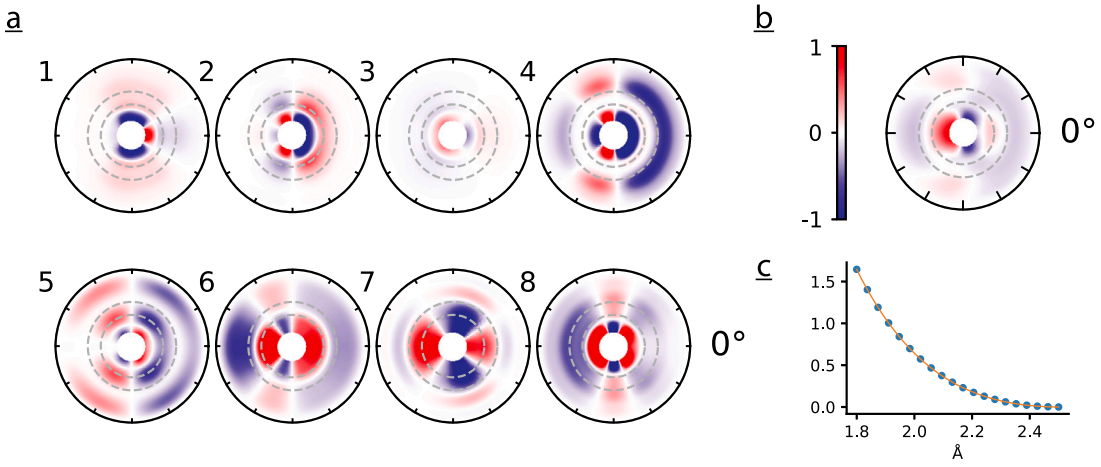

Fig. 3. Visualizations of spline filters of the "(1,8) final" model from Table 1 for atomic distances in the range [2.5 Å, 7 Å]. (a) Plots of the individual $G^{\alpha}_{3,i}$ filters, where integer labels above each plot indicate the index α. (b) The average of all $G^{\alpha}_{3,i}$ 3-body filters. (c) The short-range pair term, as described in Section 2.2. Note that the radial splines $f^{\alpha}_{c_j}$, $f^{\alpha}_{c_k}$, and $\rho^{\beta}_{c_j}$ use linear extrapolation for distances outside of the domain defined by their knots. Dashed gray lines in the polar plots correspond to the first and second nearest-neighbor distances (2.86 Å and 4.05 Å respectively) for FCC Al at room temperature [43]. Each point in the polar plots is computed by fixing atom i at the origin, fixing atom j at the given (r, θ), fixing atom k along the $\theta = 0$ axis, then integrating G^{α}_{3} over $r_k \in [1.5, 7.0]$. Note that there is a forced symmetry in the polar plots since the $g^{\alpha}_{c_{jk}}$ splines in Eq. (2) take $\cos(\theta)$ as input. All polar plots use the same color scale, where values are clipped to fall within a chosen range to optimize for visibility while avoiding information loss. Values for $r < 1.5$ Å, which was the smallest distance sampled in the Al dataset as shown in Fig. B.6, are omitted to ensure that high signals at small atomic distances would not wash out the rest of the colors in the plots. Though all plots in this figure are technically in units of eV, this would not be true if skip connections were not used.

3.2. Model visualization

A major advantage of s-NNP over many other MLIPs, especially when coupled with the modifications from Section 2.2, is that the spline filters $G^{\alpha}_{3,i}$ and $G^{\beta}_{2,i}$ lend themselves to easy visualization. This can be valuable for helping model developers and users to better understand how their model is interacting with the data or influencing simulation results. The polar plots in Fig. 3a, corresponding to the "(1,8) l = 5, int. all" model, represent the total activation of the 3-body filters induced by placing an atom at a given (r, θ). These visualizations make it easy to recognize aspects of the local environments around an atom that are learned during training to have lower energy. For example, the

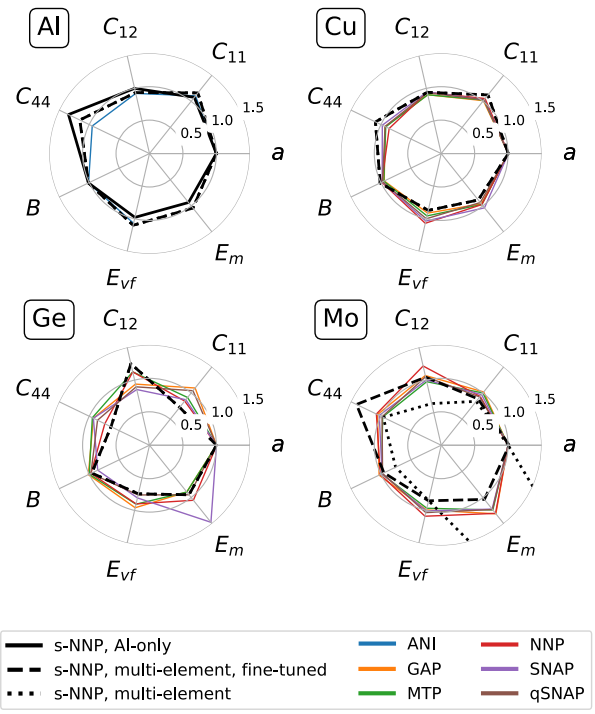

Fig. 4. Property predictions of the lattice constant (a), cubic elastic constants (C_{ij}), bulk modulus (B), and vacancy formation and migration energies (E_{vf} and E_m) for s-NNP models compared to existing MLIPs. s-NNP results are from this work, ANI results from [37], and all others from [41]. All properties have been normalized with respect to the DFT-predicted values. Note that [37] did not compute E_m for ANI. The "(1,8) $l = 5$, int. all" model (solid black line), which was trained only to Al, predicts all seven properties well, with the exception of C_{44}, which has a relatively small magnitude compared to the other elastic constants. The performance of the multi-element s-NNP model (dashed black lines) is within the range of the other single-element MLIPs, though s-NNP's predictions for Ge are somewhat distorted. The "fine-tuned" s-NNP model was manually adjusted after training was completed in order improve the property predictions, as described in Section 3.3.

averaged filter shown in Fig. 3b has an attractive behavior of the model for bond angles $60° \lesssim \theta_{jik} \lesssim 90°$ in addition to repulsion for angles larger than approximately 120°. Many of the individual filters in Fig. 3a also show characteristic features at various bond angles, especially for bond lengths near the first and second nearest neighbor distances in FCC aluminum. Fig. 3c shows the short-range pair term, which was learned to have a strongly repulsive contribution, as would be expected based on the Pauli exclusion principle.

It is worth mentioning that similar plots to the ones shown in Fig. 3 could also be generated for other NNP-like models, for example by visualizing each of the components of the vector output of the first hidden layer in an ANI model. However, most other neural network-based architectures would have significantly more filters to visualize given the size of their hidden layers. For example, the ANI model in Table 1 has 96 nodes in its first hidden layer, thus making it more difficult to interpret the results. Furthermore, since ANI (and most other models) does not use skip connections summing the hidden layer directly into the output, any resultant visualization will not necessarily be in units of energy, meaning it may undergo significant non-linear transformations as it passes through the network.

3.3. Flexibility tests

The high interpretability of the s-NNP spline filters, as discussed in Section 3.2, is particularly valuable when the filters can be applied to different chemical systems in order to enable cross-system comparisons. For example, if multiple s-NNP models using the same spline filters were trained to different chemical systems, then the networks of each model could be analyzed in order to understand how differences in chemistries result in different sensitivities to the local environment embeddings defined by the spline filters. The multi-element model trained to the datasets described in Section 2.4 uses the same spline filters for all of the data, but separate NNs for each element type. The network architecture maintained a funnel-like structure of $(12, 8, 4, 2, 1)$. It also used four additional 3-body filters in order to increase its fitting capacity, for a total of one short-range pair filter and 12 3-body filters, though without the use of skip connections or the Softplus activation function. In the case of the multi-element model, the use of skip connections would imply a "background" energy that is consistent across all four element types and is augmented by the network contributions for each element. While this assumption sounds plausible in theory, we found that in practice a model using skip connections resulted in poorer property predictions than one which did not. This decreased performance when using skip connections can likely be attributed to a lack of universality of the background energy learned by the splines, possibly due to the larger concentration of Al data dominating the fitting and causing the filters to learn a background energy which is only applicable to Al. While it is possible that re-weighting the dataset could help alleviate this issue, it also seems likely that there is simply not a background energy that is valid for all four elements simultaneously. If this is the case, then it would be valuable to explore in future work how the s-NNP architecture could be modified to allow the model to utilize skip connections without using the *same* skip connections for each elemental network (for example, by only connecting a subset of the splines to each network's output).

The property predictions of the multi-element model plotted in Fig. 4 show that a model using shared filters for multiple datasets can learn to make reasonable property predictions for all four elements studied in this work. While the initial predictions for most of the elements were relatively good, the cubic elastic constants and bulk modulus for Mo were noticeably under-predicted, and the vacancy migration energy was far too large to be considered acceptable (dotted line in Fig. 4). Due to the relatively few number of spline filters used, and their high degree of interpretability, we were able to fine-tune the model by zeroing out the network weights of hand-chosen filters in order to remove their contribution to the Mo energy predictions. For example, the contributions of each spline filter can be visualized individually in order to isolate the influence of each filter on the properties of interest (see Fig. C.8). Following this approach, we were able to improve the Mo predictions to bring them within an acceptable range without altering the predictions for the other three elements (dashed lines in Fig. 4). However, we note that the original training errors of the model (before fine-tuning) as shown in Fig. C.10 were comparable to those of the NNIPs from [41], suggesting that the property predictions of the multi-element s-NNP model may have been able to have been improved by re-balancing the dataset. Similar to what was described in Section 2.4, a possible explanation for this is that the Mo dataset did not fully constrain the portions of the spline filters relevant to computing the properties of interest, and their shapes were therefore dictated by the Al dataset, leading to corrupted Mo property predictions. Further research into methods for preventing this type of "cross-pollution" of information would be valuable for constructing more flexible and generalizable models.

In order to gain additional insights into the differences in energetics between the Al, Cu, Ge, and Mo systems, we compute the sensitivities of the energies predicted by the multi-element model with respect to each of the 3-body spline filters. The sensitivity of the total energy, E, for all configurations N with respect to a given 3-body spline filter G_3^α can be computed as:

$$s_E^\alpha = \sum_{i=1}^{N} |\partial E_{\text{s-NNP}} / \partial G_{3,i}^\alpha| / E_{\text{DFT}}, \tag{11}$$

which can be easily computed via back-propagation.

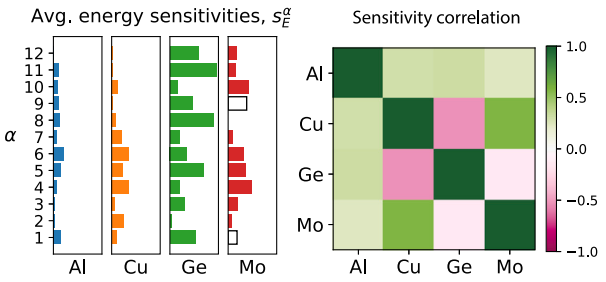

Fig. 5. Analysis of sensitivities s_E^α (Eq. (11)) of the predicted energies with respect to the 3-body spline filters, G_3^α, for each elemental model. We compute the sensitivity by calculating the derivative of the predicted energy with respect to the spline filter outputs, normalized by the DFT reference value, then summing over all atoms in the training set. Note that these sensitivities are computed with respect to the total energy E, which includes the network contributions. The left panel plots the raw values, while the right panel shows the correlation between the sensitivities of each element. The empty bars for Mo correspond to the filters which were removed from the Mo energy predictions during fine-tuning as described in Section 3.3. The corresponding spline filters are visualized in Fig. C.7.

Calculation and comparison of these sensitivities for each dataset, as shown in Fig. 5, highlight the learned similarities and differences between the four elemental datasets. Examination of the correlation between the sensitivities (right panel of Fig. 5) shows that the Cu and Mo datasets have similar filter sensitivities, while Ge is the only element to have a negative correlation with any of the other elements. The Al dataset appears to be uniformly similar to all three remaining elements, reflecting the fact that most of the geometric configurations of the Cu/Ge/Mo datasets are well-sampled by the Al dataset (see Fig. C.9). We hypothesize that the negative correlation of the Ge sensitivities is due to an attempt by the model to encode the large cluster of outlying Ge points in the UMAP visualizations shown in Fig. C.9.

4. Discussion

4.1. Understanding model behavior

The main purpose behind the modifications proposed in Fig. 2 and Section 2.2 is to simplify the process of understanding the behavior of s-NNP models. This not only improves the usefulness of visualizations like those shown in Figs. 3 and C.7, but can also aid in debugging model predictions in practice. In this section, we will discuss each modification from Section 2.2 in greater detail in order to highlight how they improve the interpretability of the model.

The first of these modifications is the use of skip connections, which result in strong theoretical changes that can not only lead to improved trainability as seen in Table 1 and other deep learning applications [44], but also greatly increase the interpretability of the model. Modifying $\vec{G}_i$ to have units of energy seemingly contrasts with the notion of the atomic "density", n_i, from Eq. (1) and the local environment descriptor, $\vec{D}_i$, from Eq. (4), both of which are considered to be intermediate representations which will only become energies once they have been transformed by a regression function (U_{c_i} or N_{c_i} respectively). However, we emphasize that these three quantities (n_i, $\vec{D}_i$, and $\vec{G}_i$) remain closely related even when $\vec{G}_i$ is in units of energy while the others are not. Although local environment descriptors are usually seen as quantifying the geometry within a local neighborhood, there is no reason that the environment descriptor may not itself also be an energy. An energy-based descriptor could then be thought of as a type of energy partitioning scheme, where in the case of s-NNP with skip connections the atomic energies contribute both linearly to the total energy (through the skip connection) and non-linearly (through the network). In fact, in our previous work [45] we observed that U_{c_i} was often learned to be a nearly linear function, meaning that it essentially served the purpose of a simple unit conversion for n_i. This linearity is essential to model interpretability, and is the main motivator for the use of skip connections in s-NNP, as it is a step towards simplifying the process of understanding how the spline filters contribute to the total energy.

Nevertheless, care must still be taken when interpreting the filter visualizations when the other modifications discussed in Section 2.2 are not also employed. For example, skip connections alone do *not* guarantee that a negative filter value means E_i will also be negative, since the network contribution $N(\vec{G}_i)$ may outweigh the skip connection term. Similarly, adjusting the filter outputs (e.g., by tuning the knot positions) may have unexpected results due to the non-linear behavior of the network. These issues are also present with all non energy-based descriptors like n_i and $\vec{D}_i$, but can be further improved upon using the remaining techniques discussed in this section.

In order to further facilitate straightforward interpretation of spline visualizations, we also choose to pin the knots of the radial splines to be 0 at the cutoff distance r_c, and we remove bias terms from the layers of the NN. As a result of this, both $N(\vec{G}_i)$ and $E_{\text{sc},i}$ will smoothly decay to 0 at the cutoff distance, thus incorporating the desired behavior which is enforced in the Behler–Parrinello NNP using the cutoff function v_c in Eq. (6). In order to account for the fact that some datasets may have non-zero energy at the cutoff distance, an external bias term, b is added so that the atomic energy converges to b at r_c. In this sense, b corresponds to the learned energy of the isolated atom. We note that this external bias term is preferable to using internal network bias or un-pinning the radial splines because it ensures that $N(\vec{G}_i)$ and $E_{\text{sc},i}$ behave similarly as r_{ij} approaches r_c.

Wrapping the network output $N_{\text{no-bias}}(\vec{G}_i)$ in the Softplus activation function guarantees that the NN contributions to the total energy are strictly non-negative. This ensures that any attractive behavior of the model arises solely from the spline filters through the skip connection term, $E_{\text{sc},i}$, thus helping to isolate certain model behaviors to specific parameter sets. It is important to note, however, that the Softplus activation should only be used in conjunction with skip connections in order to ensure that the model can predict negative energies.

A common challenge for many MLIPs is ensuring a repulsive behavior at small values of r_{ij}. This difficulty is not reflected in classical potentials, however, as classical models often include explicit repulsive terms. While a data-driven solution to this problem is possible, by explicitly adding short-range dimers into the training dataset, many MLIP developers choose instead to adjust the form of their IP. One such method used in the literature is to augment the model by introducing an auxiliary potential designed to capture the repulsive behavior of a pair potential. For example, in [46] a "composite potential" was constructed by first fitting a repulsive "auxiliary potential" to DFT data, then fitting a "main potential" to the residuals of the pair potential using a Gaussian Approximation Potential [2]. We incorporate a similar idea by including a short-range 2-body filter, $G_{2,i}$, which is summed directly into the total energy without ever passing through the NN. Note that this is different from the skip connections, which are summed directly into the energy *and* passed through the NN. Having a spline filter which only contributes linearly to the total energy means that it is free of any of the interpretability issues described for the spline filters which are used as input to the NN. While this short-range pair term does not necessarily guarantee a repulsive behavior at small distances, we observe empirically that it is often learned during training to have a strongly repulsive shape. A similar technique is used by the SNAP interatomic potential [47], where the repulsive ZBL pair potential is added to the potential form as a "reference potential".

We note that the auxiliary potential from [46] (or the short-range 2-body filter used here) and the skip connections serve related purposes: to have a linear portion of the model in the hope that it will serve as a type of interpretable background energy to which the non-linear part of the model can then apply a correction. While the possible existence of a background energy is relatively easy to believe (take, for example, a simple repulsive potential that decays at large atomic

distances), it is not obvious that the correction term should be a *function* of this background energy, as is assumed to be the case with the skip connections used here. In fact, it is much more intuitive that the background energy and non-linear correction terms be entirely decoupled from each other—such a model could be implemented within the s-NNP framework by summing a subset of the spline filters into the output energy without also passing them through the network. Although we experimented with such an architecture, we found that this type of "decoupled" model resulted in higher errors than the optimal model used in this work. However, these results are not sufficient on their own to draw any significant conclusions about the validity of coupling/decoupling the background energy and correction term, and we believe that this should be explored further in future work.

4.2. s-NNP's relation to other models

The s-NNP architecture shown in Fig. 1 can be further understood by drawing relationships between itself and other models from the literature, particularly the UF3 model [48] and the original Behler–Parrinello NNP [1]. s-NNP can be compared to UF3 and NNP by analyzing the differences between the two key components of each model: the embedding function for encoding local atomic environments into a descriptor, and the regression function for mapping the descriptor into an energy. Although it can be difficult to clearly distinguish between the embedding/regression portions of most MLIPs due to the inability to definitively establish the roles of all parameters in a deep model, we will attempt to break each model down in intuitive ways to facilitate comparison. One can view the vector $\vec{G}_i$ from Eq. (9) as a descriptor generated by an embedding function defined by the spline filters $G^{\alpha}_{3,i}$ and $G^{\beta}_{2,i}$. This embedding technique is most similar to the UF3 method, which also decomposes the energy into two-body and three-body terms described by spline basis functions. Though there are some differences in the exact details of the UF3 and s-NNP spline functions, for example UF3's use of tricubic B-splines instead of the 1D cubic Hermite splines of s-NNP, the general principle is the same. While s-NNP's embedding function is most similar to that of UF3, its regression function is identical to that of the Behler–Parrinello NNP [1]. Therefore, in order to help analyze the performance of s-NNP with respect to other models in the literature, it is suitable to think of s-NNP as a combination of a UF3-like embedding function with an NNP regressor. Or, equivalently, as an NNP using a spline-based embedding function instead of atom-centered symmetry functions [1]. However, neither UF3 nor NNP utilize all of the interpretability improvements discussed in Section 2.2.

The recently-proposed EAM-R model [49] is the most closely related model to s-NNP in the literature, as it also incorporates components of both a classical model (EAM) and an MLIP (an NNP). EAM-R is a composite potential (using the terminology described in Section 4.1) where the auxiliary potential is an EAM model, and the main potential is an RANN (an NNP with descriptors inspired by the analytical MEAM equations [31]). Similar to this work, the developers of EAM-R observed that combining a classical model with an MLIP resulted in both improved stability relative to an MLIP and improved accuracy to a classical potential alone. Despite s-NNP's similarity to EAM-R, s-NNP has some key differences, namely the use of spline filters (instead of analytical MEAM-inspired descriptors) and some of the interpretability improvements discussed in Section 2.2. In particular, the spline filters may be expected to be more flexible than the RANN descriptors (similar to how s-MEAM is more flexible than analytical MEAM) for a given computational cost, and benefit from the ability to enforce smoothness and convergence through curvature penalties and knot pinning. Furthermore, s-NNP's use of skip connections and the removal of the internal network bias greatly improve the interpretability of the model, as discussed in Section 2.2. Although EAM-R does not include these modifications, it is an excellent example of an MLIP which could easily incorporate these same interpretability improvements.

4.3. Computational costs

The computational cost of s-NNP is dominated by the evaluation of $\vec{G}_i$, and is particularly dependent upon the choice of N_3. In fact, basic profiling tests revealed that the filter evaluations accounted for approximately 95% of the total CPU and GPU time. This behavior can be understood heuristically by the fact that Eq. (2) involves approximately $\mathcal{O}(N_n^2)$ spline evaluations for each filter where N_n is the average number of neighbors within the cutoff distance (due to the summation over triplets of atoms), as opposed to the relatively few matrix multiplications associated with the evaluation of the neural network. In general, the computational cost of inference with an s-NNP potential will scale sub-linearly with N_3 (some speedup can be achieved by performing batched spline evaluations for the filters in Eq. (2)). An important practical implication of this is that in order to improve the accuracy of a given s-NNP model (and other NNP-based MLIPs as well), it is much more computationally efficient to increase the size of the network rather than the size of the embedding function. On the other hand, increasing N_3 may lead to larger increases in accuracy (up to a point) than what is achievable by only increasing the network size

Given the performances of the s-NNP models in Table 1 and the timing comparisons observed in our previous work between s-MEAM and a NNP [45], it may be expected that an s-NNP could be constructed that achieves identical errors to ANI while maintaining a higher speed. The "(1,8) $l = 5$, int. all" model is already nearing this threshold, especially taking into account that the values for ANI reported in Table 1 use ensemble-averages over 8 models, as reported by the original authors [37], which they say can make the energy and force errors "20% and 40% smaller, respectively" and may account for the differences in performance as compared to "(1,8) $l = 5$, int. all". However, given that the major difference between ANI and s-NNP is in their choice of descriptors (an ACSF variant, as opposed to spline embeddings), further research analyzing the relative amount of information that can be embedded within each descriptor would be valuable to help explain the discrepancies in errors.

5. Conclusion

In this work we developed a novel framework using spline-based filters coupled with a neural network regressor in order to blend the strengths of both classical and ML IPs. We use this framework to probe the gap between these two classes of models, observing performance limits of linear ("classical") models that can be overcome using even small neural networks. We then show that this improved performance can be maintained while incorporating architectural changes which improve the interpretability of the model, such as the use of skip connections, an external bias term, a Softplus activation function, and a short-range pair term. Finally, we demonstrate that the information-rich filter layer can be used as a reference point for performing cross-system analyses, and correlate well enough with model behavior to enable manual tuning to improve property predictions. Future studies applying the visualization techniques shown here to practical applications, and exploring methods of isolating contributions of specific elements to subsets of the model parameters would be valuable for continuing to refine the interpretability improvements explored in this work. Furthermore, efforts building upon this work could continue to improve the s-NNP design by incorporating equivariance or a message-passing network, which may lead to better performance and improved scaling of model size with number of elements.

CRediT authorship contribution statement

Joshua A. Vita: Methodology, Software, Validation, Formal analysis, Visualization, Writing – original draft. **Dallas R. Trinkle:** Writing – review & editing, Supervision, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This research was supported by the National Science Foundation through awards NSF/NRT-1922758 and NSF/HDR-1940303. Computational resources were provided on the HAL computing cluster [50] by the National Center for Supercomputing Applications.

Code and data availability

The code used for training s-NNP models can be found at https://github.com/TrinkleGroup/snnp. All datasets can be obtained from their original sources.

Appendix A. Training details

All s-NNP models in this work were trained using the code provided at https://github.com/TrinkleGroup/snnp on a single GPU from the HAL computing cluster [50]. The energy and force terms in the loss function were given weights of 10 and 1 respectively. L2 regularization was applied to the network parameters with a weight of 0.01. The AMSGrad variant of the Adam optimizer was used, with an initial learning rate of 0.001 and a MultiStepLR scheduler. Inner and outer cutoff radii of 2.5 and 7.0 were used, as specified in the main text.

Appendix B. Distributions of r_{ij} and θ_{jik}

Fig. B.6 shows that the Al dataset well samples the range of r_{ij} and $\cos\theta_{jik}$ values present in the Cu, Ge, and Mo datasets. Notably, the Al dataset has a much more uniform sampling of both of these values, likely due to both its large size and more diverse sampling technique (active learning). These results suggest that the Al dataset may help to constrain regions of the 2-body and 3-body splines that are under-sampled by the other three datasets. The inner cutoff radius of 2.5 Å(below which the short-range pair term becomes non-zero) was chosen to be slightly larger than the smallest atomic distances sampled by the datasets in order to ensure that the inner knots of the splines would be well constrained by the data.

Appendix C. Multi-element model

Fig. C.7 shows the spline filter visualizations corresponding to the 12 splines used by the multi-element model. As described in Section 3.3, the multi-element model does not use skip connections, which means that the filters in Fig. C.7 are not necessarily in units of energy and may be drastically transformed before being mapped into the output of the model. Although this greatly reduces the utility of the spline visualizations, the filters can still be understood in the context of the model sensitivities reported in Fig. 5. It can be seen that filters 1 and 9 (which were removed during the manual fine-tuning process) both have negative activations for short bond lengths and bond angles less than 30°. The filters can be further analyzed by plotting their activations over a range of lattice constants, as shown in Fig. C.8, which reveals that filters 1 and 9 both yield strongly repulsive contributions for small lattice constants. The removal of these filters for the Mo predictions drastically lowered the predicted E_m, which is consistent with our intuition that removal of filters 1 and 9 should result in a softer potential.

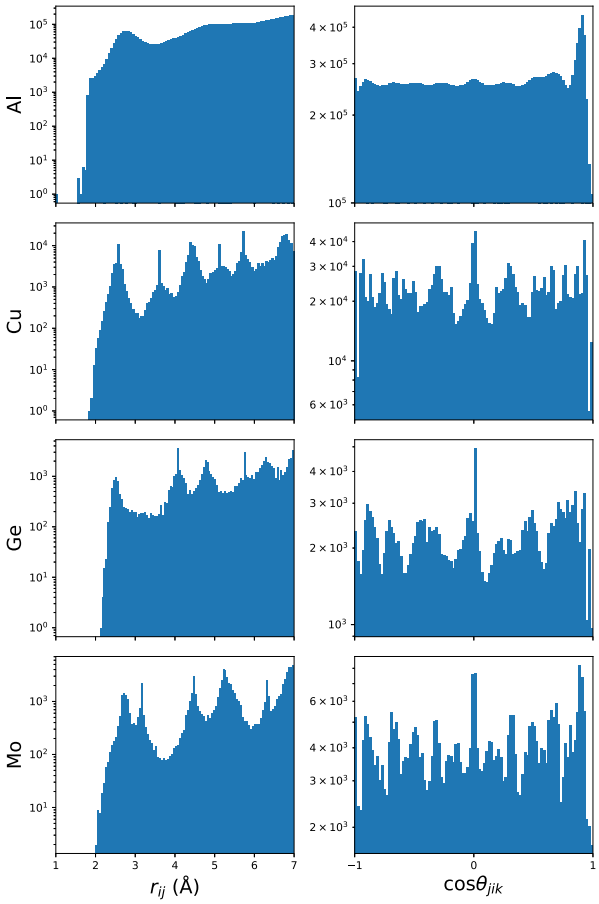

Fig. B.6. Histograms of r_{ij} and $\cos\theta_{jik}$ values sampled from the Al, Cu, Ge, and Mo datasets used in this work.

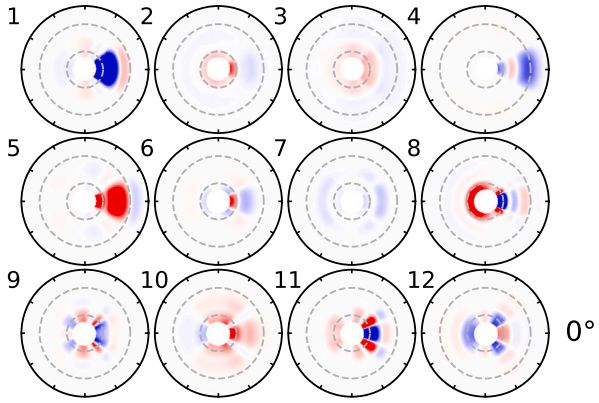

Fig. C.7. Polar plot visualizations of the 3-body spline filters for the multi-element s-NNP. Gray dashed lines mark 2 Å and 5 Å for reference.

Fig. C.9 supports the expectation from Appendix B that the Al dataset encompasses a majority of the data from the Cu/Ge/Mo datasets. As can be seen from Fig. C.9, this statement appears to be generally true, with the exceptions of some small outlying clusters of Cu and Ge points. The small isolated clusters surrounding the main

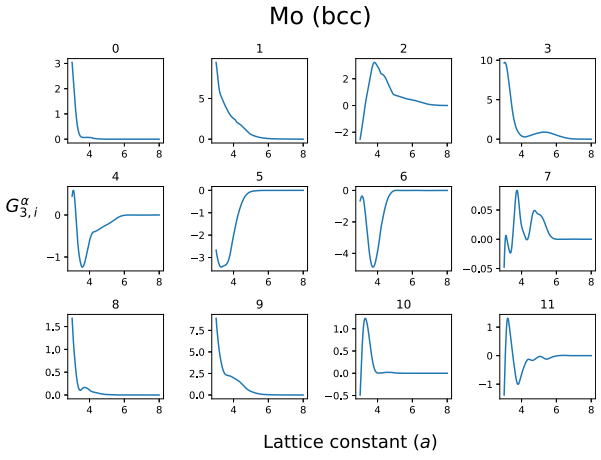

Fig. C.8. Spline filter activations, $G^{\alpha}_{3,i}$, for the multi-element model from Section 3.3 on the E vs. a curve for BCC Mo. Filter numbers 1 and 9 were removed from the Mo property predictions during the fine-tuning process by zeroing out the corresponding network weights as described in Section 3.3.

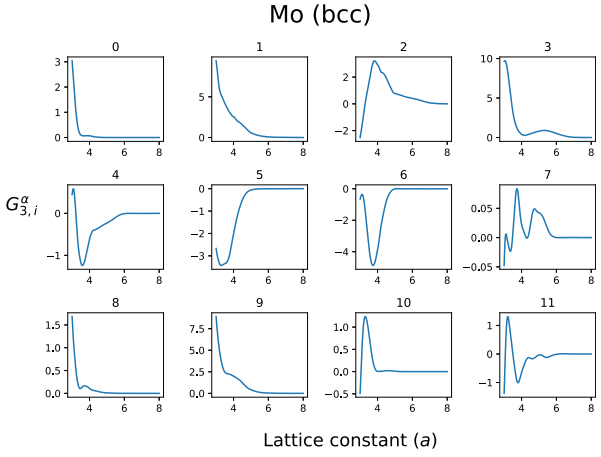

Fig. C.9. UMAP plots of the vectors $\vec{G}_i$ generated by applying the spline filters from the multi-element s-NNP model from Section 3.3, colored by element type.

manifold correspond to the strained configurations from the Cu/Ge/Mo datasets, while the larger outlying cluster of Ge points correspond to a subset of the Ge configurations which were sampled by low- and high-temperature MD simulations.

Fig. C.10 provides a breakdown of the errors of the multi-element model (prior to fine-tuning) for each elemental datasets. The apparent contradiction between the low RMSE values (compared to those from [41]) shown in Fig. C.10 and the unacceptably high errors in property predictions for Mo reported in Fig. 4 suggests that the configurations in the Mo dataset were not sufficiently representative of the properties of interest. While it is possible that the property predictions of the multi-element s-NNP may have been able to be improved if even lower Mo RMSE values could have been obtained, the fact that many models with similar errors [41,45] had good property predictions suggests some kind of deficiency in the dataset which would require further analysis in order understand fully.

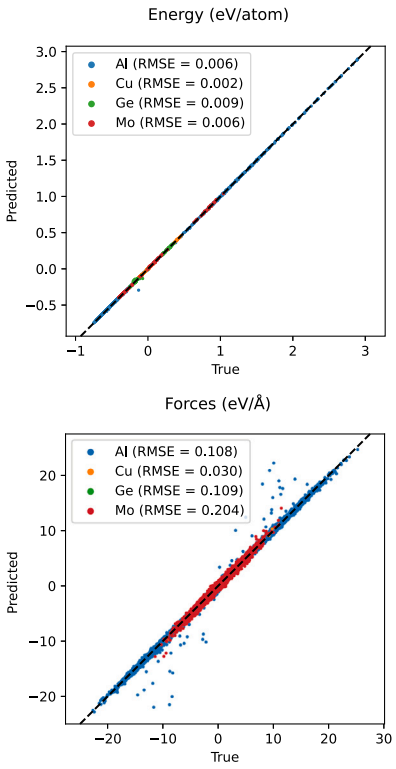

Fig. C.10. Parity plots for energy and force predictions of the multi-element model described in Section 3.3 before manual fine-tuning of the model. The computed RMSE values match the expected errors for NN-based MLIPs on the Cu/Ge/Mo dataset based on the results from [41], though the s-NNP model here uses a larger cutoff distance. Note that the RMSE value for Al should not be compared directly to those in Table 1, since the results here use only the 20% of the Al dataset as described in Section 2.4.

References

[1] Jörg Behler, Michele Parrinello, Generalized neural-network representation of high-dimensional potential-energy surfaces, Phys. Rev. Lett. 98 (14) (2007) http://dx.doi.org/10.1103/physrevlett.98.146401.

[2] Albert P. Bartók, Mike C. Payne, Risi Kondor, Gábor Csányi, Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons, Phys. Rev. Lett. 104 (2010) 136403, http://dx.doi.org/10.1103/PhysRevLett.104.136403.

[3] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, Boris Kozinsky, E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials, Nature Commun. (ISSN: 2041-1723) 13 (1) (2022) 2453, http://dx.doi.org/10.1038/s41467-022-29939-5, URL https://www.nature.com/articles/s41467-022-29939-5.

[4] A.P Thompson, L.P Swiler, C.R Trott, S.M Foiles, G.J. Tucker, Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials, J. Comput. Phys. 285 (2015) 316–330, http://dx.doi.org/10.1016/j.jcp.2014.12.018.

[5] Alexander V. Shapeev, Moment tensor potentials: A class of systematically improvable interatomic potentials, Multiscale Model. Simul. 14 (3) (2016) 1153–1173, http://dx.doi.org/10.1137/15m1054183.

[6] Ralf Drautz, Atomic cluster expansion for accurate and transferable interatomic potentials, Phys. Rev. B 99 (1) (2019) http://dx.doi.org/10.1103/physrevb.99.014104.

[7] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, George E. Dahl, Neural message passing for quantum chemistry, 2017, URL http://arxiv.org/abs/1704.01212 arXiv:1704.01212.

[8] K.T Schütt, H.E Sauceda, P.-J Kindermans, A Tkatchenko, K.-R. Müller, SchNet – A deep learning architecture for molecules and materials, J. Chem. Phys. (ISSN: 0021-9606) 148 (24) (2018) 241722, http://dx.doi.org/10.1063/1.5019779, URL http://aip.scitation.org/doi/10.1063/1.5019779.

[9] Ilyes Batatia, Simon Batzner, Dávid Péter Kovács, Albert Musaelian, Gregor N.C. Simm, Ralf Drautz, Christoph Ortner, Boris Kozinsky, Gábor Csányi, The design space of e(3)-equivariant atom-centered interatomic potentials, arXiv (2022) URL http://arxiv.org/abs/2205.06643.

[10] Tim Mueller, Alberto Hernandez, Chuhong Wang, Machine learning for interatomic potential models, J. Chem. Phys. (ISSN: 0021-9606) 152 (5) (2020) 050902, http://dx.doi.org/10.1063/1.5126336, URL http://aip.scitation.org/doi/10.1063/1.5126336.

[11] Sergei Manzhos, Tucker Carrington, Neural Network Potential Energy Surfaces for Small Molecules and Reactions, Chem. Rev. (ISSN: 0009-2665) 121 (16) (2021) 10187–10217, http://dx.doi.org/10.1021/acs.chemrev.0c00665, https://pubs.acs.org/doi/10.1021/acs.chemrev.0c00665.

[12] Anders S. Christensen, Lars A. Bratholm, Felix A. Faber, O. Anatole von Lilienfeld, FCHL revisited: Faster and more accurate quantum machine learning, J. Chem. Phys. (ISSN: 0021-9606) 152 (4) (2020) 044107, http://dx.doi.org/10.1063/1.5126701, URL http://aip.scitation.org/doi/10.1063/1.5126701.

[13] Albert Musaelian, Simon Batzner, Anders Johansson, Lixin Sun, Cameron J. Owen, Mordechai Kornbluth, Boris Kozinsky, Learning local equivariant representations for large-scale atomistic dynamics, 2022, URL http://arxiv.org/abs/2204.05249, arXiv:2204.05249.

[14] Johannes Gasteiger, Florian Becker, Stephan Günnemann, GemNet: Universal directional graph neural networks for molecules, 2021, URL http://arxiv.org/abs/2106.08903 arXiv:2106.08903.

[15] Mojtaba Haghighatlari, Jie Li, Xingyi Guan, Oufan Zhang, Akshaya Das, Christopher J. Stein, Farnaz Heidar-Zadeh, Meili Liu, Martin Head-Gordon, Luke Bertels, Hongxia Hao, Itai Leven, Teresa Head-Gordon, NewtonNet: A Newtonian message passing network for deep learning of interatomic potentials and forces, 2021, URL http://arxiv.org/abs/2108.02913. arXiv:2108.02913.

[16] Nicholas Lubbers, Justin S. Smith, Kipton Barros, Hierarchical modeling of molecular energies using a deep neural network, J. Chem. Phys. (ISSN: 0021-9606) 148 (24) (2018) 241715, http://dx.doi.org/10.1063/1.5011181, URL http://aip.scitation.org/doi/10.1063/1.5011181.

[17] Weihua Hu, Muhammed Shuaibi, Abhishek Das, Siddharth Goyal, Anuroop Sriram, Jure Leskovec, Devi Parikh, C. Lawrence Zitnick, ForceNet: A graph neural network for large-scale quantum calculations, 2021, arXiv:2103.01436.

[18] Ilyes Batatia, Simon Batzner, Dávid.Péter Kovács, Albert Musaelian, Gregor N.C. Simm, Ralf Drautz, Christoph Ortner, Boris Kozinsky, Gábor Csányi, The design space of E(3)-equivariant atom-centered interatomic potentials, 2022, URL http://arxiv.org/abs/2205.06643, arXiv:2205.06643.

[19] Gabriele C. Sosso, Ji Chen, Stephen J. Cox, Martin Fitzner, Philipp Pedevilla, Andrea Zen, Angelos Michaelides, Crystal nucleation in liquids: Open questions and future challenges in molecular dynamics simulations, Chem. Rev. 116 (12) (2016) 7078–7116, http://dx.doi.org/10.1021/acs.chemrev.5b00744.

[20] R Ravelo, T.C Germann, O Guerrero, Q An, B.L. Holian, Shock-induced plasticity in tantalum single crystals: Interatomic potentials and large-scale molecular-dynamics simulations, Phys. Rev. B 88 (13) (2013) http://dx.doi.org/10.1103/physrevb.88.134101.

[21] Jürg Diemand, Raymond Angélil, Kyoko K. Tanaka, Hidekazu Tanaka, Large scale molecular dynamics simulations of homogeneous nucleation, J. Chem. Phys. 139 (7) (2013) 074309, http://dx.doi.org/10.1063/1.4818639.

[22] James C. Phillips, David J. Hardy, Julio D.C. Maia, John E. Stone, João V. Ribeiro, Rafael C. Bernardi, Ronak Buch, Giacomo Fiorin, Jérôme Hénin, Wei Jiang, Ryan McGreevy, Marcelo C.R. Melo, Brian K. Radak, Robert D. Skeel, Abhishek Singharoy, Yi Wang, Benoît Roux, Aleksei Aksimentiev, Zaida Luthey-Schulten, Laxmikant V. Kalé, Klaus Schulten, Christophe Chipot, Emad Tajkhorshid, Scalable molecular dynamics on CPU and GPU architectures with NAMD, J. Chem. Phys. 153 (4) (2020) 044130, http://dx.doi.org/10.1063/5.0014475.

[23] Luis A. Zepeda-Ruiz, Alexander Stukowski, Tomas Oppelstrup, Vasily V. Bulatov, Probing the limits of metal plasticity with molecular dynamics simulations, Nature 550 (7677) (2017) 492–495, http://dx.doi.org/10.1038/nature23472.

[24] J.E. Jones, On the determination of molecular fields. —II. from the equation of state of a gas, Proc. R. Soc. Lond. Ser. A 106 (738) (1924) 463–477, http://dx.doi.org/10.1098/rspa.1924.0082.

[25] Murray S. Daw, M.I. Baskes, Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals, Phys. Rev. B 29 (12) (1984) 6443–6453, http://dx.doi.org/10.1103/physrevb.29.6443.

[26] R.A. Buckingham, The classical equation of state of gaseous helium, neon and argon, Proc. R. Soc. Lond. Ser. A 168 (933) (1938) 264–283, http://dx.doi.org/10.1098/rspa.1938.0173.

[27] J. Tersoff, New empirical model for the structural properties of silicon, Phys. Rev. Lett. 56 (6) (1986) 632–635, http://dx.doi.org/10.1103/physrevlett.56.632.

[28] Donald W Brenner, Olga A Shenderova, Judith A Harrison, Steven J Stuart, Boris Ni, Susan B. Sinnott, A second-generation reactive empirical bond order (REBO) potential energy expression for hydrocarbons, J. Phys.: Condens. Matter 14 (4) (2002) 783–802, http://dx.doi.org/10.1088/0953-8984/14/4/312.

[29] Tzu-Ray Shan, Bryce D. Devine, Travis W. Kemper, Susan B. Sinnott, Simon R. Phillpot, Charge-optimized many-body potential for the hafnium/hafnium oxide system, Phys. Rev. B 81 (12) (2010) http://dx.doi.org/10.1103/physrevb.81.125328.

[30] Adri C.T. van Duin, Siddharth Dasgupta, Francois Lorant, William A. Goddard, ReaxFF: a reactive force field for hydrocarbons, J. Phys. Chem. A 105 (41) (2001) 9396–9409, http://dx.doi.org/10.1021/jp004368u.

[31] M.I. Baskes, Modified embedded-atom potentials for cubic materials and impurities, Phys. Rev. B 46 (5) (1992) 2727–2742, http://dx.doi.org/10.1103/physrevb.46.2727.

[32] T.J Lenosky, B Sadigh, E Alonso, V.V Bulatov, T.D de la Rubia, J Kim, A.F Voter, J.D. Kress, Highly optimized empirical potential model of silicon, Modelling Simul. Mater. Sci. Eng. 8 (6) (2000) 825–841, http://dx.doi.org/10.1088/0965-0393/8/6/305.

[33] Kurt Hornik, Maxwell Stinchcombe, Halbert White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1989) 359–366, http://dx.doi.org/10.1016/0893-6080(89)90020-8.

[34] J.S Smith, O Isayev, A.E. Roitberg, ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost, Chem. Sci. 8 (4) (2017) 3192–3203, http://dx.doi.org/10.1039/c6sc05720a.

[35] Frank H. Stillinger, Thomas A. Weber, Computer simulation of local order in condensed phases of silicon, Phys. Rev. B 31 (8) (1985) 5262–5271, http://dx.doi.org/10.1103/physrevb.31.5262.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, 2015, URL https://arxiv.org/abs/1512.03385.

[37] Justin S. Smith, Benjamin Nebgen, Nithin Mathew, Jie Chen, Nicholas Lubbers, Leonid Burakovsky, Sergei Tretiak, Hai Ah Nam, Timothy Germann, Saryu Fensin, Kipton Barros, Automated discovery of a robust interatomic potential for aluminum, Nature Commun. 12 (1) (2021) http://dx.doi.org/10.1038/s41467-021-21376-0.

[38] Justin S. Smith, Ben Nebgen, Nicholas Lubbers, Olexandr Isayev, Adrian E. Roitberg, Less is more: Sampling chemical space with active learning, J. Chem. Phys. 148 (24) (2018) 241733, http://dx.doi.org/10.1063/1.5023802.

[39] Justin S. Smith, Benjamin T. Nebgen, Roman Zubatyuk, Nicholas Lubbers, Christian Devereux, Kipton Barros, Sergei Tretiak, Olexandr Isayev, Adrian E. Roitberg, Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning, Nature Commun. 10 (1) (2019) http://dx.doi.org/10.1038/s41467-019-10827-4.

[40] Atomistic-ml/ani-al repository, 2010, https://github.com/atomistic-ml/ani-al, Accessed: 2010-09-30.

[41] Yunxing Zuo, Chi Chen, Xiangguo Li, Zhi Deng, Yiming Chen, Jörg Behler, Gábor Csányi, Alexander V. Shapeev, Aidan P. Thompson, Mitchell A. Wood, Shyue Ping Ong, Performance and cost assessment of machine learning interatomic potentials, J. Phys. Chem. A 124 (4) (2020) 731–745, http://dx.doi.org/10.1021/acs.jpca.9b08723.

[42] Materialsvirtuallab/mlearn repository, 2010, https://github.com/materialsvirtuallab/mlearn, Accessed: 2010-09-30.

[43] Philip N.H. Nakashima, The crystallography of aluminum and its alloys, 2020, http://dx.doi.org/10.48550/ARXIV.2002.01562, URL https://arxiv.org/abs/2002.01562.

[44] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, Tom Goldstein, Visualizing the loss landscape of neural nets, 2017, URL https://arxiv.org/abs/1712.09913.

[45] Joshua A. Vita, Dallas R. Trinkle, Exploring the necessary complexity of interatomic potentials, Comput. Mater. Sci. 200 (2021) 110752, http://dx.doi.org/10.1016/j.commatsci.2021.110752.

[46] Diego Milardovich, Dominic Waldhoer, Markus Jech, Al-Moatasem Bellah El-Sayed, Tibor Grasser, Building robust machine learning force fields by composite gaussian approximation potentials, Solid-State Electron. 200 (2023) 108529, http://dx.doi.org/10.1016/j.sse.2022.108529.

[47] Mitchell A. Wood, Aidan P. Thompson, Extending the accuracy of the SNAP interatomic potential form, J. Chem. Phys. 148 (24) (2018) http://dx.doi.org/10.1063/1.5017641.

[48] Stephen R. Xie, Matthias Rupp, Richard G. Hennig, Ultra-fast interpretable machine-learning potentials, 2021, URL https://arxiv.org/abs/2110.00624.

[49] Mashroor S. Nitol, Khanh Dang, Saryu J. Fensin, Michael I. Baskes, Doyl E. Dickel, Christopher D. Barrett, Hybrid interatomic potential for sn, Phys. Rev. Mater. 7 (4) (2023) http://dx.doi.org/10.1103/physrevmaterials.7.043601.

[50] Volodymyr Kindratenko, Dawei Mu, Yan Zhan, John Maloney, Sayed.Hadi Hashemi, Benjamin Rabe, Ke Xu, Roy Campbell, Jian Peng, William. Gropp, HAL: Computer system for scalable deep learning, in: Practice and Experience in Advanced Research Computing, ACM, 2020, http://dx.doi.org/10.1145/3311790.3396649.