Feature selection for high-dimensional neural network potentials with the adaptive group lasso

IOP Publishing Mach. Learn.: Sci. Technol. 5 (2024) 025043 https://doi.org/10.1088/2632-2153/ad450e

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
26 December 2023

REVISED
28 February 2024

ACCEPTED FOR PUBLICATION
29 April 2024

PUBLISHED
17 May 2024

PAPER

Feature selection for high-dimensional neural network potentials with the adaptive group lasso

Johannes Sandberg[1,2,3,*], **Thomas Voigtmann**[1,2], **Emilie Devijver**[4] and **Noel Jakse**[3]

[1] Institut für Materialphysik im Weltraum, Deutsches Zentrum für Luft- und Raumfahrt (DLR), 51170 Köln, Germany
[2] Department of Physics, Heinrich-Heine-Universität Düsseldorf, Universitätsstraße 1, 40225 Düsseldorf, Germany
[3] Université Grenoble Alpes, CNRS, Grenoble INP, SIMaP, F-38000 Grenoble, France
[4] Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France
[*] Author to whom any correspondence should be addressed.

E-mail: johannes.sandberg@grenoble-inp.fr

Keywords: feature selection, neural network potentials, adaptive group lasso, molecular dynamics, aluminium, Lennard Jones, boron

Supplementary material for this article is available online

Abstract

Neural network potentials are a powerful tool for atomistic simulations, allowing to accurately reproduce *ab initio* potential energy surfaces with computational performance approaching classical force fields. A central component of such potentials is the transformation of atomic positions into a set of atomic features in a most efficient and informative way. In this work, a feature selection method is introduced for high dimensional neural network potentials, based on the adaptive group lasso (AGL) approach. It is shown that the use of an embedded method, taking into account the interplay between features and their action in the estimator, is necessary to optimize the number of features. The method's efficiency is tested on three different monoatomic systems, including Lennard–Jones as a simple test case, Aluminium as a system characterized by predominantly radial interactions, and Boron as representative of a system with strongly directional components in the interactions. The AGL is compared with unsupervised filter methods and found to perform consistently better in reducing the number of features needed to reproduce the reference simulation data at a similar level of accuracy as the starting feature set. In particular, our results show the importance of taking into account model predictions in feature selection for interatomic potentials.

1. Introduction

During the last decade, machine learning interaction potentials (MLIPs) have become a commonplace method for molecular dynamics simulations in material science and chemistry [1, 2], following a broader trend of data-driven approaches in material science [3, 4]. *Ab initio* simulations, using for instance density functional theory (DFT) force calculations [5], have good accuracy and broad applicability, but suffer from poor scalability. Being trained to reproduce *ab initio* forces and energies, MLIPs were shown to combine many of the benefits of *ab initio* with the scalability and performance of classical force fields [6–8], thereby opening up new avenues of research into nucleation [9–11], structure-property relationship in alloys [12, 13], and amorphous solids [14, 15] to name a few.

A wide variety of MLIPs have been proposed, often relying on a local decomposition of the high dimensional potential energy into a sum of local contributions. Methods such as the spectral neighbor analysis potential [16] rely on a linear regression over a set of nonlinear descriptors of the local atomic environment. Nonlinear dependencies can be added by the use of kernel regression, as in the Gaussian approximation potential [17, 18], or by using neural networks (NN) as in the deep potential framework [19] and the high dimensional neural network potential [20]. More recently, methods based on graph neural networks have seen a lot of traction [21], including methods based on equivariant transformations [22]. Attempts have also been made to go beyond local interaction in what has been referred to as the third and

fourth generations of machine learned potentials [23]. For most MLIPs, it is necessary to transform the bare atomic coordinates into a set of atomic descriptors [1] describing the local environment of each atom. The purpose of this transformation is to enable a local description, ensure invariance to local symmetry transformations, and to guarantee that the input to the machine learning (ML) model is of constant dimension, even as the number of atomic neighbors can change during a simulation.

Computing the descriptors is often the main time consuming part of applying a NN potential (NNP), compared to the NN evaluation and backpropagation. As such, care is needed when designing the set of atomic features, and in particular one has to weight the need for a detailed description of the atomic environment against the additional computational cost of having a large feature space. There is also some evidence that larger feature sets can negatively impact generalization [24]. Feature selection [25] allows for a data driven way of designing such feature sets by identifying those features out of a larger collection that are the most relevant, and discarding redundant ones. The simplest approach to feature selection are filter methods. Such methods select features by looking only at the dataset, before training takes place, and are as such model independent. Imbalzano *et al* [26] proposed three such methods for use with MLIPs. Two of these are based on minimizing the Pearson correlation (PC), and maximizing the Euclidean distance, respectively between the selected features. The third one is based on the CUR decomposition [27], which can be regarded as an analogue of the singular value decomposition, constructing a low-dimensional representation of the data matrix but using only rows (columns) of the original matrix chosen such that the reconstruction error is minimized.

Filter methods can be contrasted with embedded methods, wherein the feature selection process is integrated into the training of a specific model. Such an embedded approach allows for explicitly taking into account model predictions, as well as interaction between different features [28]. A famous embedded method is the lasso [29], based on regularization using the $L1$ norm of the input parameters of a linear model. Lasso has previously been used to construct MLIPs for a variety of elements based on ridge regression [30, 31], and has been applied beyond MLIPs to predict directly material properties starting from large sets of material descriptors [32]. The latter led to the development of the SISSO method [33] in the framework of materials discovery, where features are subjected to an initial screening based on their correlation to the target property, before being further selected using the lasso, allowing for selection from more than billions of candidate material descriptors. However, as it induces sparsity at the level of individual parameters, lasso is not applicable as a feature selection method for NNPs.

While much of the focus for feature selection was traditionally on linear regression, likely owing to the nonlinear nature of NNs, recent works tried to extend methods to the nonlinear case. Methods based on the group lasso (GL) has been applied to NNs as early as 2017 [34]. It was, however, shown that this direct application of GL to NNs cannot consistently discard truly irrelevant features, a problem that can be avoided by using an adaptive penalty for an adaptive GL (AGL) approach [35]. Another recent method is LassoNet [28], adding bypass connections from each input variable to the NN output, applying a lasso penalty on the bypass weights and using them to constrain the maximum values of the input weights. This change in architecture, however, deviates from the simple networks used in most common NNP implementations, while also introducing an additional hyperparameter that in principle needs to be tuned. For these reasons the AGL might be more directly suitable for NNPs.

In this article, we introduce an approach of feature selection based on the AGL method applied to high dimensional NNPs (HDNNPs), with the aim of showing that the use of a method that takes into account the interplay between features in the specific estimator allows for better selection of atomic fingerprints. This type of NNP model is known to work well for many systems, and has been well studied, making it a natural framework for our study. While more recent graph-based models avoid the need for feature selection, such deep models have been shown to suffer from potential stability issues [36, 37]. It should be noted though that methods for inducing sparsity might still have benefits [38] More importantly, message passing poses a problem for scalability of graph-based models, due to difficulties in parallelization [39]. This is especially relevant for situations requiring large scale simulations, in which feature selection is of particular interest. We consider three different systems: *Lennard–Jones* (LJ), serving as a simple and well known generic model whose analytic expression has no explicit angular dependence; *Aluminium* (Al), which serves as a relatively simple sp bonding metal; *Boron* (B), which is known to have a particularly complex structure with a high degree of directional covalent bonding [40, 41], in addition to radial interactions. Notably the B ground state is not fully understood [40], with a crystalline structure dominated by B_{12} icosahedra, with a pronounced short range order in the liquid [42]. Taken together, these three systems provide increasingly complex, and increasingly angularly dependent, interactions. We find that for Al the AGL method is competitive with filter methods. For the other systems it is explicitly shown by example how the filters can fail to select features that are necessary, while they are discovered by our method, illustrating the advantage of an embedded feature selection approach.

The remainder of the article is as follows. Section 2 provides background on our datasets, the HDNNP approach, the AGL method, and the computational tools used. Section 3 covers the results of training HDNNPs with AGL, comparing to the CUR and PC methods, as well as simulations used to test the effect of the reduced feature sets in production. Finally, section 4 provides the main conclusions and outlook of the paper.

2. Method

2.1. Datasets

A first step of training a HDNNP is to construct a dataset of reference structures. Figure 1 illustrates the location of the thermodynamic state points included in datasets used for the training of the three systems.

The dataset for LJ was extracted from a set of LAMMPS [43] simulations of 256 atoms at temperatures ranging from 0.5 to 1.5 (LJ units), and densities 0.9 to 1.1, in both solid (fcc) and liquid configurations. We use the standard LJ pair potential, given for interatomic distance $r < r_c$ by

$$V = 4\epsilon\left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right). \tag{1}$$

All the simulations are performed with parameters $\sigma = \epsilon = 1$, particle mass $m = 1$, and cutoff radius $r_c = 2.8$. Figure 1(a) shows the thermodynamic states included in the dataset. Each thermodynamic state was sampled 1333 times, with an interval of 0.3 time units (300 timesteps), for a total of 28 000 configurations. Note that the coexistence lines in figure 1, reproduced from [44], are valid in the limit of infinite cutoff, and merely included as visual guide.

In the case of Al, our reference data is the same as in our previous article [9]. This dataset consists of 24 300 configurations extracted from DFT-based *Ab Initio* molecular dynamics (AIMD) simulations performed in VASP [45] using the LDA functional [46] in an augmented plane wave framework with a cutoff of 241 eV. Configurations in the dataset cover fcc, bcc, and hcp crystalline states, and the liquid, at a variety of temperatures and pressures the details of which we refer to the original article [9]. Figure 1(b) shows the thermodynamic points sampled to construct the dataset. Liquid states, and fcc crystals at ambient pressure were sampled 1000 times each. The remaining crystal states were each sampled 100 times.

For B, we extract reference configurations from the AIMD trajectories used in [42], complemented with additional simulations for α-rhombohedral, α-tetragonal, and β-rhombohedral crystals at temperatures ranging from 10 K to 2000 K in steps of 200 K, extracted from the *Materials Project* database [47]. Additional high-pressure simulations were also included, to probe the short-range interaction. Figure 1(c) shows the thermodynamic state of each simulation trajectory, with the number of configurations drawn from it. Each trajectory was sampled with an interval of 45 fs (30 timesteps), for a total of 45 000 configurations. These simulations were performed using the Perdew Wang GGA functional [48] with a 300 eV augmented plane wave cutoff sampling only the Γ point, for consistency with [42].

In all cases, the simulations were performed in an NVT ensemble with a Nosé thermostat controlling the temperature, and pressure is controlled by fixing the volume of the simulation box. To ensure sampling of equilibrium states, each trajectory was preceded by an equilibration period ranging from 500 time units for LJ, and 100 to 200 ps for Al and B.

2.2. HDNNPs

The interaction between atoms in a material is frequently described in terms of a potential, depending in principle on the positions of all atoms in the many-particle system. This interaction is often short-sighted, and can be treated as sum of atomic contributions depending only on the local structure of each atom, within an appropriate cutoff radius r_c

$$E_{\text{total}} = \sum_{i=1}^{N_{\text{atoms}}} E_i. \tag{2}$$

A HDNNP [20, 49] is constructed from this decomposition by assigning a NNP to each species of atom, mapping between the local environment and the corresponding atomic energy contribution E_i. The input to the HDNNP are the atomic positions, which are transformed into a fingerprint vector for each atom, serving as input to the atomic NNP. Training then consists of fitting the full HDNNP to the total potential energy obtained from *ab initio*. Often the derivative of the HDNNP is fitted to the *ab initio* forces as well, but for simplicity in focusing on the feature selection and following our previous work [9], we train only to the energies in this work.

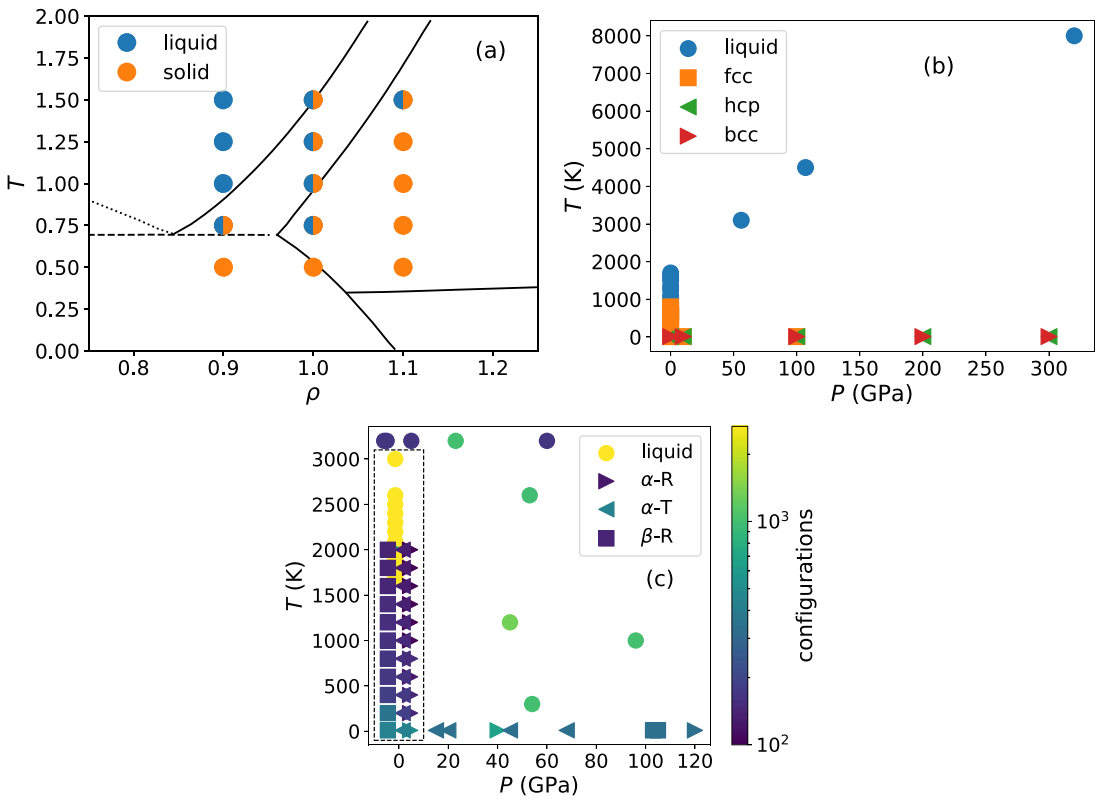

Figure 1. Thermodynamic points sampled in the construction of our datasets. Colors, symbols indicate respectively simulations started in different thermodynamic phases. (a) Temperature–density (T–ρ) phase diagram for LJ. For points with both colors, two separate simulation sets were included. (b) Temperature–pressure (T–P) phase diagram for Al. (c) Temperature–pressure (T–P) phase diagram for B. Points on the $P = 0$ line, inside the dashed rectangle, have been shifted horizontally for readability. Here, colors represent the number of states sampled.

There are many options in choosing atomic descriptors, with [1] offering a brief overview of some common types. In this work, we use the Behler-Parrinello symmetry functions (SF) [50], which is the conventional choice for HDNNPs. These consist of the radial G^2 and angular G^5 SFs defined by

$$G_i^2 = \sum_j \mathrm{e}^{-\eta\left(R_{ij}-R_s\right)^2} f_\mathrm{c}\left(R_{ij}\right) \tag{3}$$

$$G_i^5 = 2^{1-\zeta} \sum_{j,k} \left(1+\Lambda\cos\theta_{ijk}\right)^\zeta \mathrm{e}^{-\eta\left(R_{ij}^2+R_{ik}^2+R_{jk}^2\right)} f_\mathrm{c}\left(R_{ij}\right) f_\mathrm{c}\left(R_{ik}\right) f_\mathrm{c}\left(R_{jk}\right) \ . \tag{4}$$

Here, R_{ij} is the distance between atoms i and j, θ_{ijk} is the angle between atoms j and k with respect to atom i, and $f_\mathrm{c}(R_{ij})$ is defined as 0 for $R_{ij} > r_\mathrm{c}$ and for $R_{ij} < r_\mathrm{c}$ as a polynomial going smoothly to 0 at the neighborhood cutoff $R_{ij} = r_\mathrm{c}$. The parameters η, ζ, Λ, and R_s allow for defining a set of features by assigning these parameters different values. Here the initial featuresets are generated by selecting parameter values on a grid, akin to the procedures described in [24, 26], with the aim of being sensitive to a range of interatomic radii and angles. The exact SF parameter values used can be found in the supplementary material.

2.3. Feature selection

The main hindrance in applying feature selection methods based on the L1 norm to NNs is the fact that the L1 norm acts on individual weights. In a NN, several weights are associated with each feature, and so to do feature selection we need to penalize these weights as a group. The GL replaces the L1 norm with Euclidean norms over groups of parameters. As the Euclidean norm of a parameter group vanishes if and only if all those parameters vanish, this allows for selecting or discarding groups of parameters simultaneously. To select features for NNs using GL we take the groups to be the input weights of feature i, $\omega^0_{i,[:]}$, with the corresponding Euclidean norm $|\omega^0_{i,[:]}|$. During training we then optimize the objective function

$$\mathrm{obj}\left(W\right) = L\left(W\right) + \frac{\lambda}{N}\sum_{i=1}^{N} |w^0_{i,[:]}|, \tag{5}$$

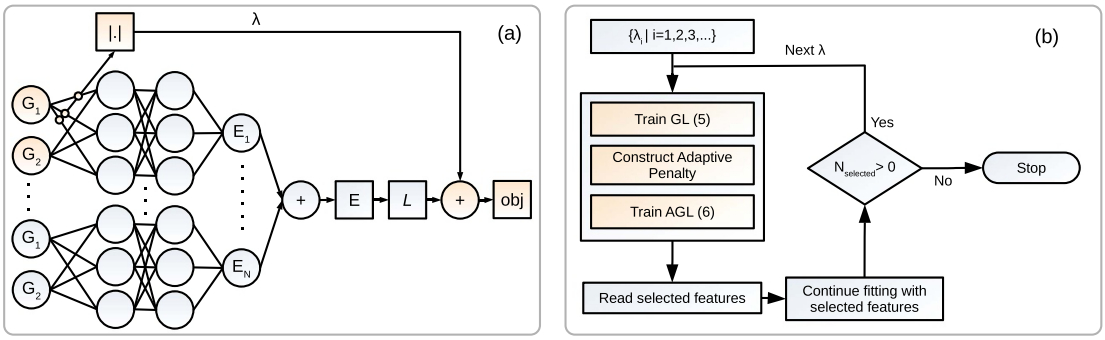

Figure 2. Schematic of feature selection for HDNNPs with AGL. (a) Illustration of a NNP, with GL penalty added to the first feature, used to construct the objective function. (b) Flowchart of the feature selection process.

with L being some loss function, in our case the mean square error, W being the weights of the neural network, N being the number of inputs, and λ being a regularization parameter used to tune the relative strength of the feature selection. A challenge in performing this optimization is the fact that the second term in (5), called the penalty, is non-smooth. In [51] a smoothed approximation of (5) is used, but here the non-smooth optimization problem is instead solved directly using a proximal gradient descent algorithm, following [52]. Figure 2(a) illustrates the GL penalty acting on one of the input features to a schematic NNP.

The adaptive version of the algorithm [35] uses a separate regularization parameter for each individual weight group. This adapted penalty is constructed from an initial training run using the non-adaptive penalty. The training is then redone with the new penalty, optimizing

$$\mathrm{obj}\,(W) = L\,(W) + \frac{\lambda}{N}\sum_{i=1}^{N}\frac{|w^0_{i,[:]}|}{|\hat{w}^0_{i,[:]}|} \tag{6}$$

with $\hat{w}^0_{i,[:]}$ being the values of $w^0_{i,[:]}$ obtained during the initial training run with the non-adaptive penalty. Depending on the value of λ, some features will have their weights go to zero during training, and can thus be discarded. This allows for selecting features by performing a search over this single parameter, following the workflow illustrated in figure 2(b).

2.4. Computational tools

Training of HDNNPs were performed using our own code, with the SF calculations being performed using N2P2 [53]. For the CUR selection we use the code implementation from [54]. Simulations with the trained potentials were performed in LAMMPS [43] using the ml-hdnnp plugin provided by N2P2. As mentioned in section 2.1 we use VASP [45] for reference *ab initio* calculations. OVITO [55] was used for some post-processing, calculating the radial distribution functions (RDFs).

3. Results and discussion

3.1. Lennard Jones system

As a first test of our method we apply the AGL to the LJ system, where the exact interactions are perfectly known. In particular, they are perfectly spherically-symmetric pair interactions, so that one might expect a feature-selection method to successfully discard features pertaining to angular directionality. The initial feature set contains 12 radial SFs, 6 of which are centered on $r_{ij} = 0$ with varying widths η, with the remaining 6 being centered on regularly spaced r_s having constant width. In addition to the radial SFs, 10 angular ones are included, using the same wide centered radial component, with varying angular width ζ in pairs of $+1$ and -1 for the Λ parameter. All the SFs use the same cutoff radius, set to the cutoff used in the reference LJ potential, $r_c = 2.8$. The NNP consists of two hidden layers with 10 neurons each.

For the feature selection, we apply the AGL method described in section 2.3 by defining a sequence of regularization parameters λ, training an initial model with the non-adaptive GL (5). This is then used to construct and retrain the model using the adaptive penalty given by (6). Each of these models has its weights randomly chosen at the beginning of the training, referred to as cold initialization, and is trained using the ADAMW optimizer [56] with learning rate set using a learning rate finder [57], and a small weight decay parameter $\gamma = 10^{-6}$ applied only to the internal weights so as to not interfere with the feature selection. The batch size was fixed at 256 configurations, and standard input normalization was used, shifting and scaling

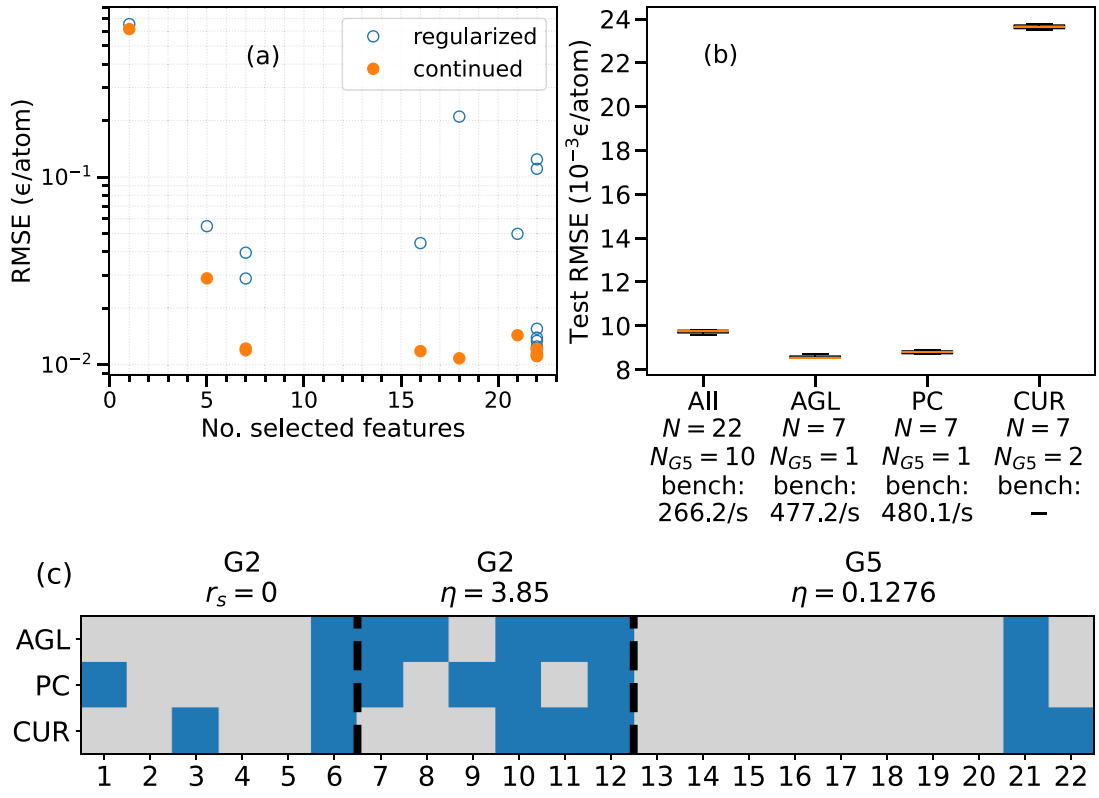

Figure 3. Selection process for LJ. (a) RMSE of models trained with different values of the regularization parameter λ, plotted against the number of selected features. Blue circles show the error at the end of training with the AGL penalty. Orange dots show the error after continuing training without penalty, with only selected features. (b) Box plot of test errors for different feature sets, with total number of features (N), number of angular features (N_{G^5}), and timesteps per second in a benchmark simulation. (c) Matrix plot of selected features. Rows correspond to different methods, with discarded features grayed out, and selected ones in blue. The features are grouped into centered G^2, shifted G^2, and G^5.

each feature to have mean 0 and standard deviation 1 over the training dataset. We let aside 10% of the training data as a hold-out validation set to monitor the model performance during training for early stopping. Crucially, for the sake of early stopping we do not monitor just the loss function, but the relevant objective function given by (5) or (6), ending training if it has not improved for 10 epochs by more than 10^{-7}. In the absence of early stopping, the training was capped at 1000 epochs for the non-adaptive part, and 10 000 during the adaptive part.

During training with the adaptive penalty, the weights corresponding to some of the inputs will vanish. Following the training for each λ we identify these weights and freeze them before continuing training without the penalty. This is to avoid the bias that is otherwise known to occur for L1 regularized models [58]. Figure 3(a) shows the validation Root RMSE for each model along this path, plotted against the number of selected features, both at the end of training with AGL (blue circles) and after continuing without the penalty (orange dots). Note that the regularization introduces a noticeable overestimation of the error associated to the selected feature sets, and so continuing the training is necessary to make an informed decision on which set of selected features to choose. In figure 3(a) one can observe an initial plateau in the lowest error reached during continued training when going from 22 selected features down to 7. We interpret this as the regime where the AGL method discards unnecessary features that lead to little decrease in performance. Going below 7 features, the model suffers a large increase in error, as the result of having to discard more and more important features.

Based on figure 3(a), we select the model with 7 features, of which 1 is of the angular type given by (4). The selected feature set is tested by training over four different random initializations, with the same training dataset, to ensure the features are not suited for just one part of the weight space. Unlike the models on the regularization path, in order to speed up convergence, these models were trained using the *cosine annealing with warm restarts* learning rate schedule [59]. With this schedule the learning rate is annealed with a cosine from a large initial value to a small value (10^{-8}) over a number of weight updates, before resetting the learning rate to its initial value and repeating the process. Here the initial period of the scheduler is set to coincide with one epoch, and to double after each reset, ending training after a total of 12 resets (8190

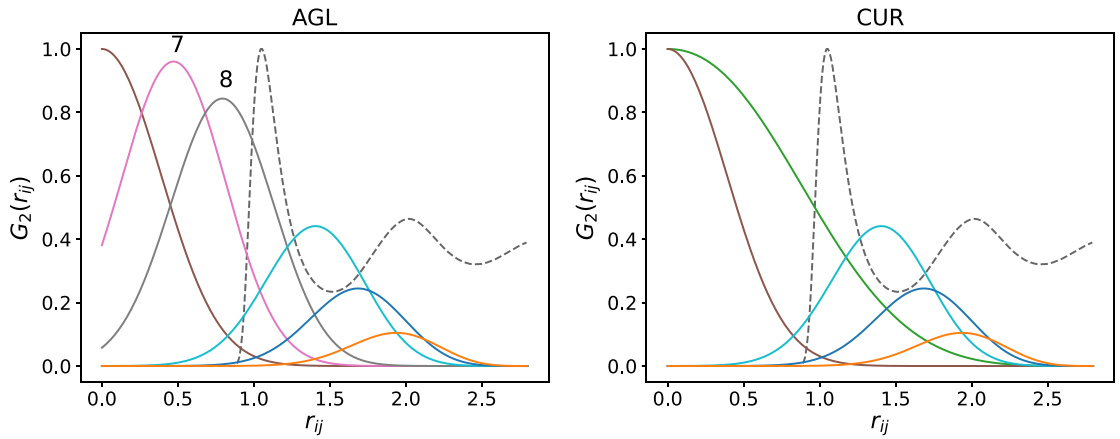

Figure 4. Radial symmetry functions (G_2) for LJ, visualized for a single atom pair, selected by AGL (left), and CUR (right). RDF of reference system included for comparison (dashed line). Features number 7 and 8 from figure 3(c) are marked.

epochs). We likewise test the starting feature set, as well as 7 features selected with the PC and CUR methods of [26]. The resulting test errors, evaluated on a held out test set, are presented in figure 3(b), together with the total number of features N and the number of angular features N_{G^5}. Additionally, we perform a benchmark simulation with each potential, consisting of 256 atoms simulated in an NVT ensemble for 6000 timesteps. These simulations ran on 48 2.7 GHz Intel Skylake cpu cores, and the average number of simulated timesteps per second of wall time is recorded and shown in figure 3(b). We note that the models trained on the features selected with CUR did not allow for a successful benchmark simulation on account of their large error, which will be discussed in more detail below.

It can be seen that there is a strong preference for radial SFs, as one would expect considering the lack of angular dependence in the reference LJ potential. Despite this, a single angular feature was selected by both the AGL and the PC filter. This is not unreasonable, since we train the LJ system with high-density configurations as reference data, where steric repulsion leads to the emergence of certain short-ranged angular order. The features selected with CUR greatly underperform those selected with the other methods, but we note that CUR performs much better for a larger number of features [26]. CUR selected two angular features, which could allow for a better reconstruction of the atomic environment overall by taking better into account the angles, but at the cost of a reduced radial resolution. As the CUR approach acts on the descriptors alone, it is largely incapable of knowing the lack of angular dependence of the energy in the ground truth. It should however be mentioned that this information could still be, to some extent, indirectly available through what configurations appear in the sampled MD trajectory used to construct the dataset.

To better illustrate the differences between the feature selection methods, we show in figure 3(c) a matrix representing the features selected by each method. The G^2 SFs selected by AGL and CUR are also plotted in figure 4, along with the RDF extracted from one of the reference simulations. Of note is that CUR discarded three consecutive shifted radial SFs in a regime where the other methods kept at least one. This raises the question of whether adding one of these SFs to the CUR features would recover a good performance. In order to test this, we create two new sets by adding to the CUR features one of the shifted radial SFs selected by AGL but discarded by CUR, marked 7 and 8 in figure 4. Adding feature number 8 reduced the test RMSE to $18.4 \times 10^{-3}\epsilon$/atom, which is a modest improvement, but still nowhere near the performance of the other sets. Instead, adding feature number 7 lowers the test RMSE to $9.40 \times 10^{-3}\epsilon$/atom, a clear indication that this is indeed a vitally important feature for this system that the CUR method failed to detect. With this feature added, the resulting model also allowed for stable simulations to be performed.

3.2. Aluminium

To test the method in a more practical setting, we turn to the case of Al. The SF parameters and network architecture is chosen as in [9]. We proceed as for LJ, training a sequence of models on increasing values of λ, using cold initialization, continuing the training after selecting the features. The resulting validation errors are plotted against the number of selected features in figure 5(a). We find 10 features to be a good compromise between few features and low error. The set is again evaluated by training a set of four models on the selected features, with different initialization, likewise for the staring features and features selected with CUR and PC. The test errors are shown in figure 5(b), along with the number of angular features selected, and number of timesteps per second in a benchmark simulation identical to the one for LJ. We see a

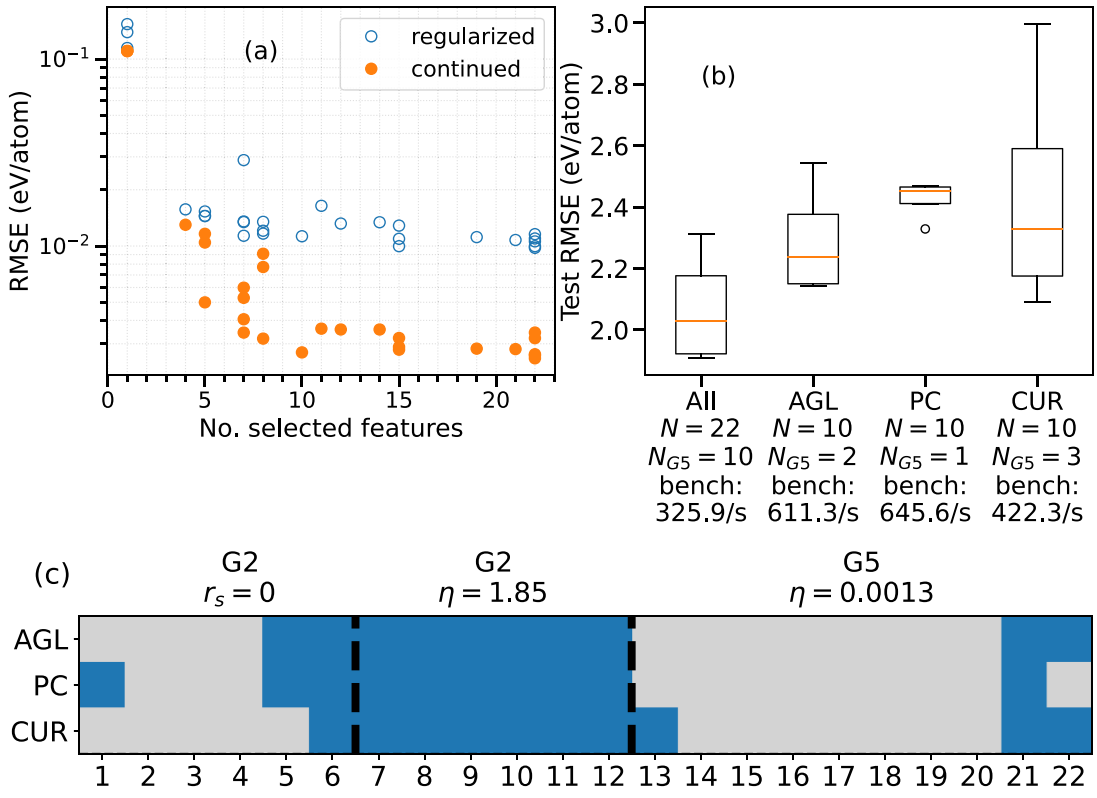

Figure 5. Selection process for Al. (a) RMSE of models trained with different values of the regularization parameter λ, plotted against the number of selected features. Blue circles show the error at the end of training with the AGL penalty. Orange dots show the error after continuing training without penalty, with only selected features. (b) Box plot of test errors for different feature sets, with total number of features (N), number of angular features (N_{G^5}), and timesteps per second in a benchmark simulation. (c) Matrix plot of selected features. Rows correspond to different methods, with discarded features grayed out, and selected ones in blue. The features are grouped into centered G^2, shifted G^2, and G^5.

significant increase in computational speed for the feature-selected potential, at a relatively small increase in error. For this system, CUR and PC seem to perform equivalently. In particular the CUR features perform much better than in the LJ case, presumably because it is asked to select more features and so the method is not forced to compromise on the radial resolution. The features selected with AGL, on average, outperform those chosen by the filters, although there is not a large difference in this case, especially considering the deviations.

A point should be made regarding the nonlinear scaling of the benchmark performance in figure 5, with respect to the number of features. This is a direct consequence of the angular G^5 features involving a double sum over neighbor atoms, as opposed to the single sum of the radial G^2 features. In addition, depending on the SF parameters, some factors appear in the calculation of several different features [53], allowing for optimizations that further complicate the scaling.

The feature sets are visualized in figure 5(c). We observe, somewhat different from the LJ case, a great overlap between the methods, and presumably the one or two features that differ between each set are not enough to cause a significant difference in the test error. In particular we notice that each model selected each shifted radial SF. Feature number 6 in figure 5(c), being also selected by each model, is identical to the shifted ones, but centered on $r_{ij} = 0$. Taken together these features can be argued to cover the entire range of interatomic distances up to the cutoff radius, allowing for a rough representation of the RDF. This preference for shifted radial SFs has also been indicated elsewhere in the literature [24].

Like in the case of LJ, there is here a preference towards radial features, with only two angular ones being chosen. We suggest a physical explanation for this preference for radial features, noting the tendency of Al to adopt a close-packed short range order and to maximize the number of nearest neighbors, due to the weakly directional sp bonding type electronic structure.

While the 10 features selected are a sensible choice, based on the training errors reported in figure 5(a), the threshold is not rigorous. From the RMSE values obtained, a selection of 8 or even only 7 features could also be argued for. In going to 7 features, a noticeable increase in the test error was observed, providing only a modest improvement in benchmark performance primarily due to an additional discarded angular SF.

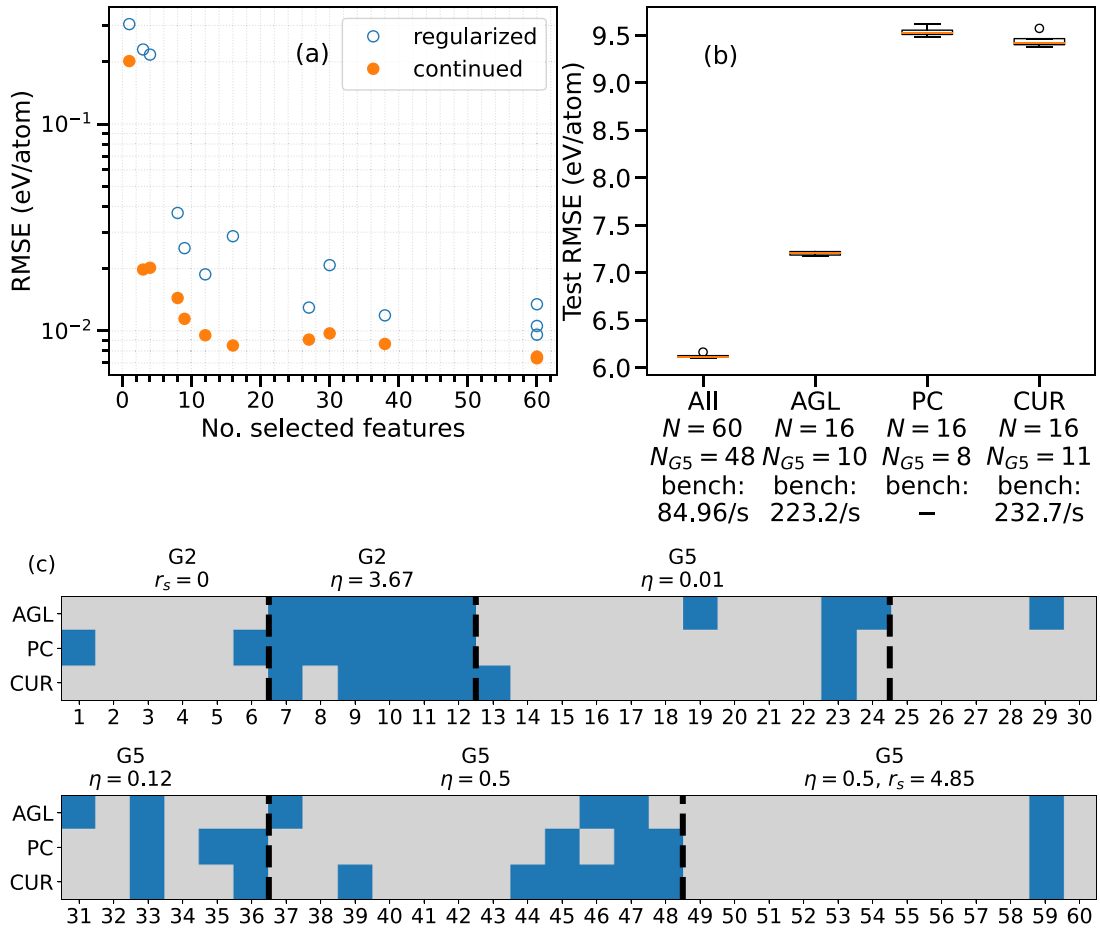

Figure 6. Selection process for B. (a) RMSE of models trained with different values of the regularization parameter λ, plotted against the number of selected features. Blue circles show the error at the end of training with the AGL penalty. Orange dots show the error after continuing training without penalty, with only selected features. (b) Box plot of test errors for different feature sets, with total number of features (N), number of angular features (N_{G^5}), and timesteps per second in a benchmark simulation. (c) Matrix plot of selected features. Rows correspond to different methods, with discarded features grayed out, and selected ones in blue.

Simultaneously, the CUR features show a significant reduction in performance, reminiscent of what was observed for LJ. In the present case, this was presumably due to the deselection of both features number 6 and 7 by CUR. Figures for these featuresets can be found in the supplementary material.

3.3. Boron

We turn now to boron as a stringent test system. Due to the complicated structure of boron, induced by strong covalent directional bonding [40–42], we expect this to be a significantly more difficult task, and to require a more complex set of features compared to Al and LJ. For our initial set of descriptors we use a set of 12 radial SFs, and 48 angular SFs, with a cutoff of 5.3 Å corresponding roughly to the outer edge of the third neighbor shell. This relatively wide cutoff was chosen in order to hopefully be able to more adequately take into account the medium-range structure known to appear in boron, primarily the open icosahedra and the bonds between them [42]. Furthermore, to allow for a potentially more complex mapping we use a larger network than for LJ and Al, with two layers of 25 hidden nodes each, providing a slight improvement in error compared to smaller network sizes.

As for the previous systems, figure 6(a) shows the validation RMSE as a function of the selected features. In this case the best-performing model, apart from the one with the full set of features, is for 16 features. We select these 16 features, and again train a set of four models to test, with the results shown in figure 6(b). In this case we not only selected a larger number of features, but the majority of features selected were of the angular type. Unlike in the previous cases, we also observe an inability of the filter methods to adequately select features for this system, with a significant increase in error for the sets selected with PC and CUR. In fact, we were unable to perform even a benchmark simulation using the models trained on the PC set, with the simulations becoming unstable. For the AGL set there is a noticeable increase in the error compared to

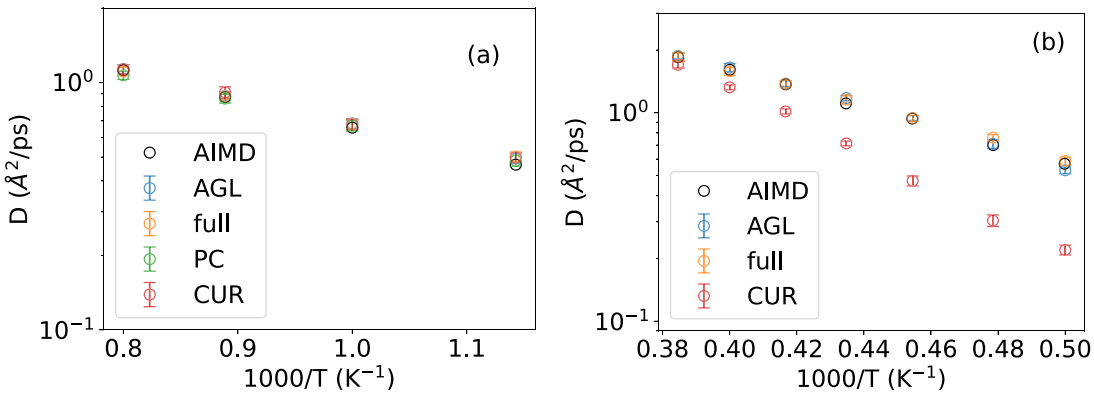

Figure 7. Arrhenius curves for the diffusion constant against temperature, with different feature sets, for (a) Aluminium and (b) Boron. Reference *ab initio* diffusion data from [42, 61].

the full set of features, but this comes with a significant improvement in the computational performance of the potential. It should be noted that the error of even the baseline HDNNP trained with all the available features is noticeable, comparing to, for instance, the errors seen in figure 5 for Al. This very likely hints at the BP SFs not being well suited for the B system, a point which has been made previously in the literature [60], highlighting the need to explore different types of fingerprints for HDNNPs. Nevertheless, the success of the AGL method in even this setting shows promise that with more well suited features the method would still be usable.

3.4. Validation of the MLIP models

While looking at the RMSE of the models on a held-out set of test configuration is useful, the true test of the quality of a MLIP is in simulations and the accurate prediction of physical quantities. For each set of features we pick out the model with the best test error and perform an NVT simulation, aiming to obtain the diffusion constant for comparison to *ab initio*. We specifically focus on the real systems, Al and B, leaving the LJ case for the supplementary material. Each simulation uses a box of 256 atoms, in order to match the finite size effect in the reference systems. For Al we prepare the system in an fcc crystal configuration, and melt it at 1250 K. The resulting liquid is then repeatedly quenched in steps of 125 K, followed by 30 ps of equilibration, down to 875 K, relatively deep in the undercooled regime. At each temperature a measurement is then performed over 1 ns of simulation at constant temperature. For B the same procedure is followed, but starting from a 2600 K liquid configuration drawn from *ab initio*, and quenching in steps of 100 K down to 2000 K. In both cases, the diffusion is calculated from the mean square displacement, *via* the Einstein relations, and averaged over a set of ten independent simulation runs. Figure 7 show the diffusion as a function of inverse temperature, for the different feature sets. In the case of Al (a), we see a good agreement across all temperatures, with none of the feature sets being obviously worse. This is not the case for B (b), where the full feature set and the features selected by AGL both agree well with *ab initio*, but the set selected by CUR show a significant deviation. For the PC set we were unable to perform a stable simulation for B, although we cannot rule out that it is possible to still train a functioning potential on these features; the CUR model has a comparable error, and in a previous iteration was also unstable.

From these simulations we also extract the RDF, shown in figure 8. One point that should be stressed here is that our aim is to evaluate the feature selection, rather than how well any of the models reproduce the AIMD reference system results. For the Al case we observe very little difference between the different NNP models, as both the initial large feature set and also the reduced sets following feature selection reproduce the AIMD results fairly well. The same holds for the LJ results, in the supplementary material. In the case of B, already the initial large feature set turns out to be not powerful enough to reproduce the boron RDF faithfully. But the feature selection by AGL does not deteriorate the agreement further, indicating that no significant performance is lost—the feature selection can be only as good as the initial starting point. This is also in contrast with the model trained on the CUR features, which is seen to greatly underperform the other two models, to an increasing extent at lower temperatures. The failure to reproduce the AIMD RDF emphasizes that boron is a challenging system for the training based on Behler-Parrinello SFs and potential energies as targets. Irrespective of this, the agreement with the AIMD MSD is very good also for the reduced feature set. We rationalize this as a result of the dynamics in boron being not predominantly determined by the radial structure encoded in the angle-averaged RDF. This additionally points to the possibility of the

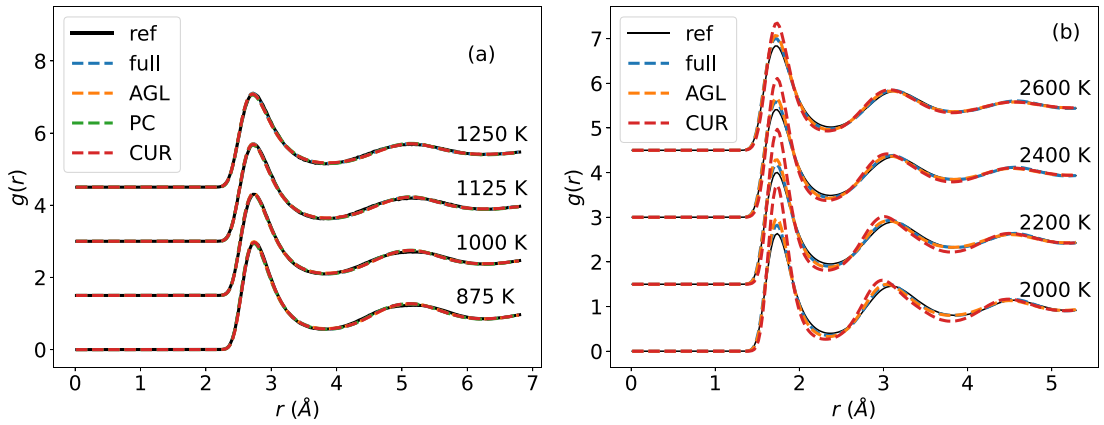

Figure 8. RDFs for different feature sets for Al (a) and B (b), at various temperatures. Plots are shifted upwards by 1.5 with increasing temperatures.

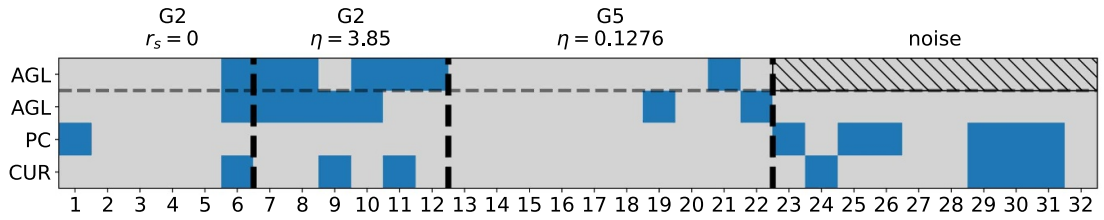

Figure 9. Matrix plot of selected features for LJ, with random noise features. Rows correspond to different methods, with discarded features grayed out, and selected ones in blue. Top row, above the dashed line, is from figure 3(c), included for reference.

standard BP SFs being not well suited for this system. In the supplementary material we present a comparison of simulation results for a B model trained with N2P2, including forces in the training data. That model yielded even worse results than did the models presented here, indicating that this is also not a problem that can be solved just by introducing forces into the training.

3.5. Confounding features

Filter methods such as PC and CUR aim to reduce the number of features by looking for subsets that minimize the overlap between those features that are kept. However, this makes them potentially vulnerable to confounding features that are uncorrelated to the relevant input, but by themselves irrelevant. This requires the initial selection of features one starts with to be carefully chosen, in order to minimize irrelevant input. However, in a system with complex structure this might not be obvious to achieve. We demonstrate in the following, that AGL performs much better in the presence of irrelevant input.

For this purpose we return to the LJ system, modifying the starting featureset by adding 10 new features consisting of random noise drawn independently from a set of Gaussian distributions, with means and variances chosen to mimic those of the real features. We note that these fake features were sampled once for each atom and configuration, and as such the values do not vary between epochs. To ensure these fake features are nonnegative, like the real ones, we only work with their absolute values. While the situation considered here is a rather implausible one to occur in a practical setting, where features are unlikely to be truly uncorrelated to the potential energy, it could potentially have implications in situations where there is noise in the training dataset.

Having nothing to do with the real data generating process, these features are truly independent from the other features as well as the target energy. Ideally these features should be discarded, but as they are independent from the real features as well as each other, we expect that neither the PC nor CUR should be able to correctly discard them. This is indeed the case, as illustrated in figure 9, showing the features selected by AGL, PC, and CUR, as well as for comparison, the set selected by AGL in the absence of fakes. The PC method clearly did not succeed, as beyond the manually selected feature it only picked out fake features. With CUR we selected some real features, indicating that the method might be more robust compared to the PC in this regard, but still it selected more fakes than real features. In contrast to the filters, the AGL managed to discard the fakes, and select a set of features. An interesting observation is that the set selected by the AGL

is slightly different to that selected in the absence of fakes. In fact, the error obtained on this set was 6.97×10^{-3}, below that of the set selected in absence of fakes. This is reminiscent of ML methods where the deliberate addition of noise helps increasing the performance in training.

4. Conclusion and outlook

We have applied the AGL as an embedded feature selection method for choosing atomic features in HDNNPs. This allows for selecting features as part of the training process, taking into account the action of the features in the resulting potential during the selection. In order to evaluate the method we have compared it to previously used unsupervised filter methods that take only into account the features themselves, aiming to minimize redundancy in the description of the local atomic environment. We find that for three test systems, ranging from a simple LJ system, to the highly complicated and directional boron system, that the AGL manages to perform as good as, or better than the other methods. This we consider the main outcome of this work. By utilizing a method that takes into account the NNP predictions, we can reduce the number of atomic features further than methods taking only into account the features themselves.

While we have applied our method to training on only energies, the next step would be to apply the method to the more common setting of fitting also forces during training. A natural question in this case is whether the inclusion of forces changes the features that are selected. It would also be a natural direction to use the method for different types of descriptors. Although the BP SFs are largely in use, and have seen plenty of success, since their introduction many other alternative descriptors have been developed. This is especially relevant considering the difficulty of even our full set of features to better reproduce the overall properties of boron, which could be an indication that the SFs are not ideally suited for this system. One can further consider multicomponent systems for which feature selection using AGL might potentially counteract the combinatorial increase in the number of features seen by traditional SF approaches. In view of recent concerns regarding the stability of MLIPs [36], it would also be interesting to study the extent to which input dimensionality affects the stability of models, and whether this can be alleviated by careful feature selection, or indeed regularization in general.

Data availability statement

The data cannot be made publicly available upon publication because they are not available in a format that is sufficiently accessible or reusable by other researchers. The data that support the findings of this study are available upon reasonable request from the authors.

The code used in this article is available at https://github.com/JohannesSandberg/HDNNP-AGL.

Acknowledgments

We acknowledge the CINES and IDRIS under Project No. INP2227/72914, as well as CIMENT/GRICAD for computational resources. We acknowledge financial support under the French-German Project PRCI ANR-DFG SOLIMAT (ANR-22-CE92-0079-01). This work has been partially supported by MIAI@Grenoble Alpes (ANR-19-P3IA-0003). J S acknowledges funding from the German Academic Exchange Service through DLR-DAAD fellowship Grant Number 509. We thank Gerhard Jung for suggesting tests with random features.

ORCID iDs

Johannes Sandberg https://orcid.org/0000-0002-0263-0255
Noel Jakse https://orcid.org/0000-0002-4031-0965

References

[1] Behler J 2016 Perspective: Machine learning potentials for atomistic simulations *J. Chem. Phys.* **145** 170901
[2] Kocer E, Ko T W and Behler J 2022 Neural network potentials: a concise overview of methods *Annu. Rev. Phys. Chem.* **73** 163–86
[3] Schmidt J, Marques M R G, Botti S and Marques M A L 2019 Recent advances and applications of machine learning in solid-state materials science *npj Comput. Mater.* **5** 83
[4] Choudhary K *et al* 2022 Recent advances and applications of deep learning methods in materials science *npj Comput. Mater.* **8** 59
[5] Hafner J 2008 *Ab-initio* simulations of materials using VASP: density-functional theory and beyond *J. Comput. Chem.* **29** 2044–78
[6] Daw M S and Baskes M I 1984 Embedded-atom method: derivation and application to impurities, surfaces and other defects in metals *Phys. Rev.* B **29** 6443
[7] Baskes M I 1992 Modified embedded-atom potentials for cubic materials and impurities *Phys. Rev.* B **46** 2727
[8] Van Duin A C T, Dasgupta S, Lorant F and Goddard W A 2001 ReaxFF: a reactive force field for hydrocarbons *J. Phys. Chem.* A **105** 9396–409

[9] Jakse N, Sandberg J, Granz L F, Saliou A, Jarry P, Devijver E, Voigtmann T, Horbach J and Meyer A 2022 Machine learning interatomic potentials for aluminium: application to solidification phenomena *J. Phys.: Condens. Matter* **51** 035402

[10] Piaggi P M, Weis J, Panagiotopoulos A Z, Debenedetti P G and Car R 2022 Homogeneous ice nucleation in an *ab initio* machine-learning model of water *Proc. Natl Acad. Sci.* **119** e2207294119

[11] Marchand D, Jain A, Glensk A and Curtin W A 2020 Machine learning for metallurgy I. A neural-network potential for Al-Cu *Phys. Rev. Mater.* **4** 103601

[12] Jain A C P, Marchand D, Glensk A, Ceriotti M and Curtin W A 2021 Machine learning for metallurgy III: a neural network potential for Al-Mg-Si *Phys. Rev. Mater.* **5** 053805

[13] Marchand D and Curtin W A 2022 Machine learning for metallurgy IV: a neural network potential for Al-Cu-Mg and Al-Cu-Mg-Zn *Phys. Rev. Mater.* **6** 053803

[14] Artrith N, Urban A and Ceder G 2018 Constructing first-principles phase diagrams of amorphous Li_xSi using machine-learning-assisted sampling with an evolutionary algorithm *J. Chem. Phys.* **148** 241711

[15] Li W, Ando Y, Minamitani E and Watanabe S 2017 Study of Li atom diffusion in amorphous Li_3PO_4 with neural network potential *J. Chem. Phys.* **147** 214106

[16] Thompson A P, Swiler L P, Trott C R, Foiles S M and Tucker G J 2015 Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials *J. Comput. Phys.* **285** 316–30

[17] Bartók A P, Payne M C, Kondor R and Csányi G 2010 Gaussian approximation potentials: the accuracy of quantum mechanics, without the electrons *Phys. Rev. Lett.* **104** 136403

[18] Deringer V L, Bartók A P, Bernstein N, Wilkins D M, Ceriotti M and Csányi G 2021 Gaussian process regression for materials and molecules *Chem. Rev.* **121** 10073–141

[19] Zhang L, Han J, Wang H, Car R and E W 2018 Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics *Phys. Rev. Lett.* **120** 143001

[20] Behler J and Parrinello M 2007 Generalized neural-network representation of high-dimensional potential-energy surfaces *Phys. Rev. Lett.* **98** 146401

[21] Schütt K T, Sauceda H E, Kindermans P-J, Tkatchenko A and Müller K-R 2018 SchNet—a deep learning architecture for molecules and materials *J. Chem. Phys.* **148** 241722

[22] Batzner S, Musaelian A, Sun L, Geiger M, Mailoa J P, Kornbluth M, Molinari N, Smidt T E and Kozinsky B 2022 E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials *Nat. Commun.* **13** 2453

[23] Behler J 2021 Four generations of high-dimensional neural network potentials *Chem. Rev.* **121** 10037–72

[24] Gastegger M, Schwiedrzik L, Bittermann M, Berzsenyi F and Marquetand P 2018 wACSF—weighted atom-centered symmetry functions as descriptors in machine learning potentials *J. Chem. Phys.* **148** 241709

[25] Chandrashekar G and Sahin F 2014 A survey on feature selection methods *Comput. Electr. Eng.* **40** 16–28

[26] Imbalzano G, Anelli A, Giofré D, Klees S, Behler J and Ceriotti M 2018 Automatic selection of atomic fingerprints and reference configurations for machine-learning potentials *J. Chem. Phys.* **148** 241730

[27] Mahoney M W and Drineas P 2009 CUR matrix decompositions for improved data analysis *Proc. Natl Acad. Sci.* **106** 697–702

[28] Lemhadri I, Ruan F and Tibshirani R 2021 LassoNet: neural networks with feature sparsity *Int. Conf. on Artificial Intelligence and Statistics* (PMLR) pp 10–18

[29] Tibshirani R 1996 Regression shrinkage and selection via the lasso *J. R. Stat. Soc.* B **58** 267–88

[30] Seko A, Takahashi A and Tanaka I 2014 Sparse representation for a potential energy surface *Phys. Rev.* B **90** 024101

[31] Seko A, Takahashi A and Tanaka I 2015 First-principles interatomic potentials for ten elemental metals via compressed sensing *Phys. Rev.* B **92** 054113

[32] Ghiringhelli L M, Vybiral J, Ahmetcik E, Ouyang R, Levchenko S V, Draxl C and Scheffler M 2017 Learning physical descriptors for materials science by compressed sensing *New J. Phys.* **19** 023017

[33] Ouyang R, Curtarolo S, Ahmetcik E, Scheffler M and Ghiringhelli L M 2018 SISSO: a compressed-sensing method for identifying the best low-dimensional descriptor in an immensity of offered candidates *Phys. Rev. Mater.* **2** 083802

[34] Yuan M and Lin Y 2006 Model selection and estimation in regression with grouped variables *J. R. Stat. Soc.* B **68** 49–67

[35] Dinh V C and Ho L S 2020 Consistent feature selection for analytic deep neural networks *Advances in Neural Information Processing Systems* vol 33 pp 2420–31

[36] Fu X, Wu Z, Wang W, Xie T, Keten S, Gomez-Bombarelli R and Jaakkola T 2022 Forces are not enough: benchmark and critical evaluation for machine learning force fields with molecular simulations (arXiv:2210.07237)

[37] Stocker S, Gasteiger J, Becker F, Günnemann S and Margraf J T 2022 How robust are modern graph neural network potentials in long and hot molecular dynamics simulations? *Mach. Learn.: Sci. Technol.* **3** 045010

[38] Wen W, Wu C, Wang Y, Chen Y and Li H 2016 Learning structured sparsity in deep neural networks *Advances in Neural Information Processing Systems* vol 29

[39] Ko T W and Ong S P 2023 Recent advances and outstanding challenges for machine learning interatomic potentials *Nat. Comput. Sci.* **3** 1–3

[40] Ogitsu T, Schwegler E and Galli G 2013 β-rhombohedral boron: at the crossroads of the chemistry of boron and the physics of frustration *Chem. Rev.* **113** 3425–49

[41] Albert B and Hillebrecht H 2009 Boron: elementary challenge for experimenters and theoreticians *Angew. Chem., Int. Ed.* **48** 8640–68

[42] Jakse N and Pasturel A 2014 Interplay between the structure and dynamics in liquid and undercooled boron: an *ab initio* molecular dynamics simulation study *J. Chem. Phys.* **141** 234504

[43] Thompson A P *et al* 2022 LAMMPS-a flexible simulation tool for particle-based materials modeling at the atomic, meso and continuum scales *Comput. Phys. Commun.* **271** 108171

[44] Schultz A J and Kofke D A 2018 Comprehensive high-precision high-accuracy equation of state and coexistence properties for classical Lennard-Jones crystals and low-temperature fluid phases *J. Chem. Phys.* **149** 204508

[45] Kresse G and Furthmüller J 1996 Efficiency of *ab-initio* total energy calculations for metals and semiconductors using a plane-wave basis set *Comput. Mater. Sci.* **6** 15–50

[46] Perdew J P and Zunger A 1981 Self-interaction correction to density-functional approximations for many-electron systems *Phys. Rev.* B **23** 5048

[47] Jain A *et al* 2013 Commentary: The materials project: a materials genome approach to accelerating materials innovation *APL Mater.* **1** 011002

[48] Wang Y and Perdew J P 1991 Correlation hole of the spin-polarized electron gas, with exact small-wave-vector and high-density scaling *Phys. Rev.* B **44** 13298
[49] Behler J 2015 Constructing high-dimensional neural network potentials: a tutorial review *Int. J. Quantum Chem.* **115** 1032–50
[50] Behler J 2011 Atom-centered symmetry functions for constructing high-dimensional neural network potentials *J. Chem. Phys.* **134** 074106
[51] Zhang H, Wang J, Sun Z, Zurada J M and Pal N R 2019 Feature selection for neural networks using group lasso regularization *IEEE Trans. Knowl. Data Eng.* **32** 659–73
[52] Feng J and Simon N 2017 Sparse-input neural networks for high-dimensional nonparametric regression and classification (arXiv:1711.07592)
[53] Singraber A, Behler J and Dellago C 2019 Library-based *LAMMPS* implementation of high-dimensional neural network potentials *J. Chem. Theory Comput.* **15** 1827–40
[54] Goscinski A, Fraux G, Imbalzano G and Ceriotti M 2021 The role of feature space in atomistic learning *Mach. Learn.: Sci. Technol.* **2** 025028
[55] Stukowski A 2009 Visualization and analysis of atomistic simulation data with OVITO–the open visualization tool *Modelling Simul. Mater. Sci. Eng.* **18** 015012
[56] Loshchilov I and Hutter F 2017 Decoupled weight decay regularization (arXiv:1711.05101)
[57] Smith L N 2017 Cyclical learning rates for training neural networks *2017 IEEE Winter Conf. on Applications of Computer Vision (WACV)* (IEEE) pp 464–72
[58] Lee J D, Sun D L, Sun Y and Taylor J E 2016 Exact post-selection inference, with application to the lasso *Ann. Stat.* **44** 907–27
[59] Loshchilov I and Hutter F 2016 SGDR: stochastic gradient descent with warm restarts (arXiv:1608.03983)
[60] Huang S-D, Shang C, Kang P-L and Liu Z-P 2018 Atomic structure of boron resolved using machine learning and global sampling *Chem. Sci.* **9** 8644–55
[61] Jakse N and Pasturel A 2013 Liquid aluminum: atomic diffusion and viscosity from *ab initio* molecular dynamics *Sci. Rep.* **3** 3135