Adaptive Selection of Atomic Fingerprints for High-Dimensional Neural Network Potentials

Johannes E. Sandberg
SIMaP, CNRS, Grenoble INP
Université Grenoble Alpes
38000, Grenoble, France
johannes.sandberg@grenoble-inp.fr

Emilie Devijver
CNRS, LIG, Grenoble INP
Université Grenoble Alpes
38000, Grenoble, France

Noel Jakse
SIMaP, CNRS, Grenoble INP
Université Grenoble Alpes
38000, Grenoble, France

Thomas Voigtmann
Institut für Materialphysik im Weltraum
Deutsches Zentrum für Luft- und Raumfahrt (DLR)
51170 Köln, Germany

Abstract

Molecular dynamics simulations of solidification phenomena require accurate representations of solid and liquid phases, making classical force fields often unsuitable. On the other hand ab initio simulations are infeasible to observe rare nucleation events. Being able to recreate ab initio quality forces, at scalability and efficiency near that of classical force fields, simulation of solidification processes is a promising area of application for machine-learned interatomic force fields. In a neural network potential the choice of input features plays a vital part in its performance. Here we propose embedded feature selection, using the adaptive group lasso technique, for identifying and removing irrelevant atomic fingerprints.

1 Background

An understanding of crystal nucleation is of great importance to controlling the properties and microscopic structure of materials [1]. Being a microscopic phenomena, Molecular Dynamics (MD) is a natural framework for the study of homogeneous nucleation [2]. A problem arises, however, in the need to accurately model the interatomic interactions simultaneously in both solid and liquid phases. Classical force fields [3, 4] are fast and allow for the study of very large systems containing up to several millions of atoms, but they are often inaccurate and lacking in transferability. In contrast ab initio simulations [5], based on density functional theory (DFT) [6], allow for a much more accurate description, and can be applied to any phase of matter and any combination of elements, but at a much higher computational cost, and limited to systems of merely a few hundred atoms. Nucleation events, however, necessitate long simulations of large systems [2].

Machine learning has found many applications within material science [7], being used for a variety of tasks such as identification of atomic structures [8], prediction of material properties [9, 10], among others. The use of machine-learned interatomic potentials (MLIPs), trained via supervised approaches, allows for bridging the gap between classical force fields and ab initio methods. By training a MLIP on data from ab initio simulations it is possible to obtain force predictions at ab initio accuracy, at a computational performance approaching that of classical force fields. This opens up new possibilities to study homogeneous nucleation at the atomic scale.

A crucial step in developing a MLIP in a High-Dimensional Neural Network Potential (HDNNP) setting [11] is to choose appropriate atomic descriptors [12] to represent accurately atomic environments

Machine Learning and the Physical Sciences workshop, NeurIPS 2022.

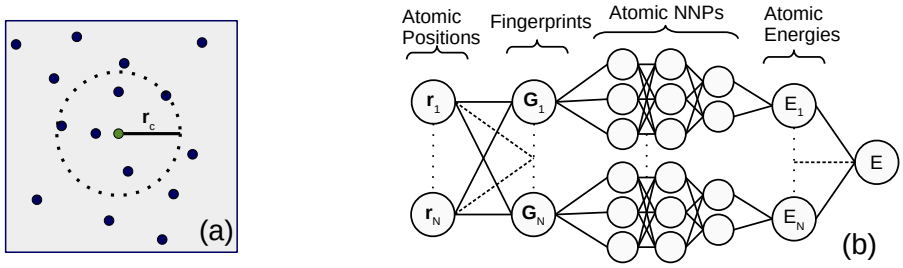

Figure 1: The High-Dimensional Neural Network Potential approach. a) Illustration of a local neighborhood of radius r_c around a central atom. b) Illustration of the HDNNP architecture.

through a set of features (atomic fingerprints). Feature selection offers a solution to optimize this choice. Previous works [13] have utilized filter methods, which do not explicitly take into account model predictions. This is in contrast to wrapper methods, such as naive forward selection [14], genetic algorithms [15], among others, and embedded methods such as the LASSO [16], and SISSO [17]. Here the objective is to develop an Adaptive Group Lasso (AGL) technique, allowing to select features during training. To our knowledge our work is the first to apply embedded feature selection to the HDNNP architecture, and to the design of MLIPs for MD simulations.

2 Method

The energy of an atom in a material typically depends on its environment within a few neighbor atom shells, within some cutoff r_c illustrated in Figure 1a. It is therefore natural to write the total energy as a sum of local atomic contributions

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

From this decomposition, a HDNNP is composed of a sum of N_{atoms} NNPs associating each local environment with E_i, as in Figure 1b. The atomic NNPs are trained indirectly by fitting the HDNNP to the known total energy obtained from ab initio simulation. It can then be differentiated with respect to the atomic positions to obtain force predictions. The inputs to the HDNNP is the atomic positions, which are transformed into a fingerprint vector G_i for each atom. These are then fed into the atomic NNP to predict the atomic energies, which are summed over to obtain the total energy.

In choosing atomic descriptors there are many options, and for a brief overview of some common types, we refer to [18]. In this work we specifically use the Behler-Parrinello symmetry functions [19], which is the traditional choice for HDNNPs. Specifically we use the G^2 and G^5 symmetry functions defined by

$$G_i^2 = \sum_j e^{-\eta(R_{ij}-R_s)^2} f_c(R_{ij}) \tag{2}$$

$$G_i^5 = 2^{1-\zeta} \sum_{j,k} (1+\lambda\cos\theta_{ijk})^\zeta e^{-\eta(R_{ij}^2+R_{ik}^2+R_{jk}^2)} f_c(R_{ij}) f_c(R_{ik}) f_c(R_{jk}) \ . \tag{3}$$

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 $f_c(R_{ij})$ is defined as 0 for $R_{ij} > r_c$ and for $R_{ij} < r_c$ as a polynomial going smoothly to 0 at the neighborhood cutoff $R_{ij} = r_c$. Other parameters such as η, ζ, etc., allow for defining a set of features by assigning these parameters different values.

A well known embedded feature selection method for linear models is the LASSO [16], in which $L1$ regularization is applied to the input parameters of the model, however this approach is not immediately suitable for Neural Networks. LassoNet [20] was recently proposed, adding bypass connections for each feature, and penalizing by the $L1$ norm of the bypass weights. Another alternative is to consider the Group Lasso (GL), which groups all the input weights of each feature

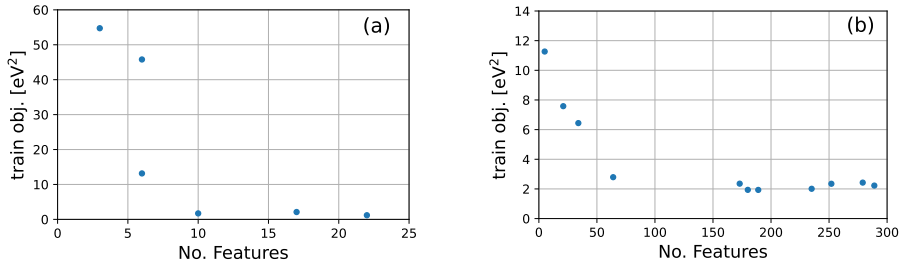

Figure 2: Validation objective for a sequence of HDNNP models, plotted against the number of selected input features obtained by varying the regularization strength λ, starting from: a) 22 hand-picked features. b) 329 features.

and regularizes with the euclidean norm of each such collection of weights. The following objective function is thus optimized:

$$\text{objective}(W) = L(W) + \frac{\lambda}{D}\sum_{i=1}^{D}|w^0_{i,[:]}|, \tag{4}$$

where W is the set of trainable weights, $L(W)$ is a suitable loss function, in our case Mean Square Error (MSE) with $L2$ regularization, $w^0_{i,[:]}$ is the vector of weights connecting input i to the first hidden layer, D is the number of features, and λ is a hyperparameter. As the euclidean norm $|w^0_{i,[:]}|$ goes to zero only if all components of $w^0_{i,[:]}$ vanish, this ensures that the input weights for each given feature is selected or discarded collectively. This objective function can be optimized using a proximal gradient descent algorithm [21], alternatively one can use a smoothed version to account for the norm being non-smooth at zero as in [22].

The basic GL approach can be further improved upon by using an adaptive penalty as in [23]. In the adaptive approach an initial training is performed using the standard GL penalty given by (4). This initial training is used to obtain an initial estimate for the weights, $\hat{w}$. With these initial estimates the training is repeated using an adaptive penalty, optimizing

$$\text{objective}(W) = L(W) + \frac{\lambda}{D}\sum_{i=1}^{D}\frac{|w^0_{i,[:]}|}{|\hat{w}^0_{i,[:]}|}. \tag{5}$$

The use of AGL over GL in NN applications is advocated by [23], where the AGL algorithms is also shown to be feature-selection consistent. We opt for AGL over LassoNet, because of adaptiveness, simpler implementation and one fewer hyperparameter. Although the AGL sometimes over-shrinks the correct features, it has been shown to be on par with LassoNet in numerical experiments[23]. This is in contrast to GL, which frequently fails to deselect insignificant features, and underperformed compared to both other methods.

3 Results

The training of HDNNPs were done using our own code, with MD simulations being performed in LAMMPS [24] making use of the ml-hdnnp plugin provided by N2P2 [25]. The dataset used to train the networks was sampled from ab initio trajectories for Aluminium, obtained using VASP [26]. In our experiments we fix our atomic NNPs to have two hidden layers of 10 nodes each, with tanh activation, focusing on Aluminium as an example system.

The feature selection process in our first setting is as follows, we pick a starting set of 22 fingerprints, identical to the ones in [27], which uses the same network architecture. These fingerprints were originally picked out by hand following the principles given in [12], and are known to be adequate, but the question is if some of them can be discarded. We train a sequence of models, with increasing values of regularization parameter λ starting from 0.005, resulting in decreasing number of input features, as depending on λ unnecessary ones will have their weights vanish. Figure 2a shows the validation objectives for such a sequence of models, plotted against the number of selected features.

Table 1: Total number of features (D), number of angular features (D_{G^5}), average test errors with standard deviation, and computational performance, for features selected from a hand-picked set.

D	D_{G^5}	MSE (eV^2)	RMSE (meV/atom)	Benchmark (timesteps/s)
22	10	0.658 ± 0.079	3.16 ± 0.193	0.211
10	3	0.702 ± 0.094	3.27 ± 0.223	0.410
6	1	1.39 ± 0.25	4.59 ± 0.42	0.581

Table 2: Total number of features (D), number of angular features (D_{G^5}), average test errors with standard deviation, and computational performance, for features selected from a large set.

D	D_{G^5}	MSE (eV^2)	RMSE (meV/atom)	Benchmark (timesteps/s)
64	11	0.250 ± 0.005	1.68 ± 0.02	0.156
32	5	0.329 ± 0.033	2.24 ± 0.11	0.267
16	2	0.329 ± 0.039	2.23 ± 0.13	0.453
8	1	0.893 ± 0.099	3.68 ± 0.20	0.579

This plot shows the typical behavior we observed, with a flat region that can be interpreted as the model being able to reduce the AGL penalty by discarding unnecessary features without increasing the loss, and sharp increases that result from the model being forced to either discard a useful feature or accept a higher AGL penalty.

We observe 10 features to be a good choice, with 6 being a possible candidate as well. For each feature set we train four models on different random dataset splits of 80% training and 20% validation data, using only the selected features. These models are evaluated on a separate test set, with the resulting average MSE and Root MSE (RMSE) being shown in table 1. We also perform, for each feature set, a short MD simulation of 1000 timesteps, using the best performing model, all starting from the same initial state drawn from a previous simulation of 10976 atoms in an undercooled liquid state. The number of simulated timesteps per second, running on a single 2.5 GHz Intel Cascade Lake 6248 cpu core, is shown in the last column of table 1. It is noteworthy that we manage to reduce the number of fingerprints by more than half, without any significant reduction in the accuracy of the potential, and almost doubling the computational speed. Further, we observed a bias towards selecting radial (G^2) features. This might be of a physical origin, noting that for Aluminium one would expect a relatively simple angular structure. As a matter of fact, Aluminium is a polyvalent metal with an s-p electronic structure, adopting a close packed short range order, maximizing the number of nearest neighbors.

In order to test the approach in a more general setting we create a large set of 269 radial fingerprints, and 60 angular ones, containing the 22 used in the previous example. Such a large set of fingerprints is unsuitable for HDNNPs in practice, with most fingerprints containing redundant information. We apply the same feature selection procedure as before, with the resulting plot of validation objective against number of features being shown in figure 2b. The first thing of note is that, although the overall structure is the same as in figure 2a, we select in general more features. Furthermore, we found there to be a large degree of correlation between the selected features. From this regularization path we select a model with 64 features, and retrain it without regularization. We also create smaller sets by picking out every other, every 4th, and every 8th feature, respectively, creating sets of 32, 16, and 8 fingerprints. Table 2 shows average test performance, and computational speed, all evaluated as previously. At a first glance the benchmarks might seem surprisingly good compared to the ones in table 1, but this will also depend on the relative number of angular features, which is lower here, as well as how much of the fingerprint calculations can be cached and reused between different fingerprints, as detailed in [25]. We note that despite the ad hoc procedure we used to shrink the 64 feature set, we still manage to outperform the models selected from the hand-picked set. Although the selection of features is stochastic, and we observe some difference in the selected features between various runs and dataset splits, we did find the selection to be fairly stable.

4 Outlook

We have applied the AGL to perform feature selection for atomic fingerprints used in HDNNPs for MD simulations of Aluminium. While we observed promising results in reducing the size of a hand-picked set of atomic fingerprints, when attempting to select from a larger set of fingerprints the method failed to reduce the size to the same extent. We hypothesize that this is due to the large degree of correlations between the fingerprints in this large set, and show that a subset of these can still outperform the features in the hand-picked set. Future work will be necessary to overcome this issue, possibly by the addition of an extra penalty term penalizing correlated features as in [28]. Additional work will be aimed at systems with different physical characteristics, such as Boron and Lennard-Jones matter, to examine the effect of physical properties on what features are selected. Further, we look to apply HDNNPs to the study of solidification in binary and multi-component systems, where the AGL could serve as a way to counteract the increased number of atomic descriptors needed to train a multi-species potential.

Acknowledgments and Disclosure of Funding

We acknowledge the CINES and IDRIS under project No. INP2227/72914, as well as CI-MENT/GRICAD for computational resources. This work was performed within the framework of the Center of Excellence of Multifunctional Architectured Materials "CEMAM" ANR-10-LABX-44-01 funded by the "Investments for the Future" program. This work has been partially supported by MIAI@Grenoble Alpes (ANR-19-P31A-0003). J. Sandberg acknowledges funding from the German Academic Exchange Service (DAAD) through the DLR-DAAD programme, grant No. 509.

References

[1] K. F. Kelton, A. L. Greer, *Nucleation in Condensed Matter: Applications in Materials and Biology*, Elsevier (2010)

[2] G. C. Sosso, et. al., *Chem. Rev.* **116**, 7078 (2016)

[3] M. I. Mendelev et al., *Phil. Mag.* **88**, 1723 (2008)

[4] B.-J. Lee, J.-H. Shim, M. I. Baskes, *Phys. Rev. B* **68**, 144112 (2003)

[5] R. Car, M. Parrinello, *Phys. Rev. Lett.* **55**, 2471 (1985)

[6] J. Hafner, *J. Comput. Chem.* **29**, 2044 (2008)

[7] J. Schmidt, et. al., *npj. comput. mat.* **5**, 83 (2019)

[8] S. Becker, E. Devijver, R. Molinier, N. Jakse, *Sci. Rep.* **12**, 3195 (2022)

[9] A. Furmanchuk, A. Agrawal, A. Choudhary, *RSC Adv.* **6**, 95246 (2016)

[10] X. Zheng, P. Zheng, R.-Z. Zhang, *Chem. Sci.* **9**, 8426 (2018)

[11] J. Behler, M. Parrinello, *Phys. Rev. Lett.* **98**, 146401 (2007)

[12] J. Behler, *Int. J. Quantum Chem.* **115**, (2015)

[13] G. Imbalzano, et. al., *J. Chem. Phys.* **148**, 241730 (2018)

[14] T. Hastie, et. al., *The elements of statistical learning: data mining, inference, and prediction*, Vol. 2, New York: springer (2009)

[15] D. A. Sofge, *Proceeding of the 2002 international conference on machine-learning and applications*

[16] R. Tibshirani, *J. Roy. Stat. Soc. B* **58**, 267 (1996)

[17] R. Ouyang, et. al., *Phys. Rev. Mater.* **2**, 083802 (2018)

[18] J. Behler, *J. Chem. Phys.* **145**, 170901 (2016)

[19] J. Behler, *J. Chem. Phys.* **134**, (2011)

[20] I. Lemhadri, F. Ruan, R. Tibshirani, *PMLR* **130**, (2021)

[21] J. Feng, N. Simon, *arXiv preprint*, arXiv:1711.07592v2 (2019)

[22] H. Zhang, J. Wang et al., *IEEE Trans. Knowl. Data. Eng.* **32**, 4 (2019)

[23] V. Dinh, L. S. T. Ho, *arXiv preprint*, arXiv:2006.00334, (2020)

[24] A. P. Thompson, et. al., *Comp. Phys. Commun.* **271**, 108171 (2022)

[25] A. Singraber, J. Behler, C. Dellago, *J. Chem. Theory Comput.* **15**, 1827 (2019)

[26] G. Kresse, J. Furthmuller, *Comput. Mater. Sci.* **6**, 15 (1996)

[27] N. Jakse, et. al., *J. Phys.: Condens. Matter* **51**, 035402 (2023)

[28] R. Chakraborty, N. R. Pal, *IEEE Trans. Neural Netw. Learn. Syst.* **26**, 35 (2015)

Broader Impact

If broadly adopted, our method, or similar ones, can reduce the computational effort of MD simulations with HDNNPs, contributing to greater energy efficiency. Reducing the number of inputs could make interpretation of the potential easier, contributing to explainability of potentials which have traditionally been regarded as black boxes. Further, more efficient simulations open up for a deeper understanding of solidification phenomena, and nucleation processes, which is a vital component of designing better materials for practical applications. For the specific case of aluminium, this strong yet lightweight material finds many practical uses in transport, wind-power generation, among others.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Details on the method are included in the text, or references. The code and data is work in progress, or still being used, and will be made available in future publications.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] Included for the benchmarks of trained HDNNPs, but not for the training of models. The latter is deemed less relevant for our discussion.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]