ARTICLE OPEN

Fast, accurate, and transferable many-body interatomic potentials by symbolic regression

Alberto Hernandez¹, Adarsh Balasubramanian¹, Fenglin Yuan¹, Simon A. M. Mason¹ and Tim Mueller^{1*}

The length and time scales of atomistic simulations are limited by the computational cost of the methods used to predict material properties. In recent years there has been great progress in the use of machine-learning algorithms to develop fast and accurate interatomic potential models, but it remains a challenge to develop models that generalize well and are fast enough to be used at extreme time and length scales. To address this challenge, we have developed a machine-learning algorithm based on symbolic regression in the form of genetic programming that is capable of discovering accurate, computationally efficient many-body potential models. The key to our approach is to explore a hypothesis space of models based on fundamental physical principles and select models within this hypothesis space based on their accuracy, speed, and simplicity. The focus on simplicity reduces the risk of overfitting the training data and increases the chances of discovering a model that generalizes well. Our algorithm was validated by rediscovering an exact Lennard-Jones potential and a Sutton-Chen embedded-atom method potential from training data generated using these models. By using training data generated from density functional theory calculations, we found potential models for elemental copper that are simple, as fast as embedded-atom models, and capable of accurately predicting properties outside of their training set. Our approach requires relatively small sets of training data, making it possible to generate training data using highly accurate methods at a reasonable computational cost. We present our approach, the forms of the discovered models, and assessments of their transferability, accuracy and speed.

npj Computational Materials (2019)5:112; <https://doi.org/10.1038/s41524-019-0249-1>

INTRODUCTION

In recent years, there have been great advances in the use of machine learning to develop interatomic potential models.^{1–20} In this approach, the development of an interatomic potential model is treated as a supervised learning problem,²¹ in which an optimization algorithm is used to search a hypothesis space of possible functions to find those that best reproduce the energies, forces, and possibly other properties of a set of training data. Potential models developed in this way are often able to achieve accuracy close to that of the method used to generate the training data, with linear scalability and orders of magnitude increase in performance. Alternatively, potential models may be generated by using fundamental physical relationships to derive a simple parameterized function. The parameters of this function are typically then fit to a smaller set of training data. Examples of potential models generated using this latter approach include the embedded-atom method (EAM) and bond-order potentials.^{22–28}

There are advantages and disadvantages to both approaches to potential model development. Machine learning can be used to develop models for a wide variety of different chemical systems, and because many machine-learning algorithms explore a large hypothesis space, they are often able to achieve very high levels of accuracy on structures where the local environments of the atoms are similar to those that are contained in the data used to train the model.^{1–3} On the other hand, models developed from fundamental physical relationships are often simpler and orders of magnitude faster than machine-learning potential models,²⁹ allowing them to be used to model systems at much longer time and length scales. As they are derived from physics, they can be expected to perform relatively well when they encounter local environments that are unlike the ones they were trained on. The

hypothesis space of these potential models is relatively small compared to most machine-learning potentials, meaning that less data is required to train them but also that they are typically unable to achieve the same level of accuracy as many potentials generated using machine learning.

Here, we present a hybrid approach in which machine learning is used to develop simple, fast potential models. Previous work has demonstrated that genetic algorithms can be effectively used to find parameters for interatomic potential models with known functional forms.^{30–35} Our approach adds to these efforts by identifying new functional forms for the models themselves. To accomplish this we use symbolic regression as implemented using genetic programming, in which simple expressions for the potential energy surface are optimized by simulating the process of natural selection.^{36,37} Genetic programming has been used to rediscover fundamental physical laws³⁸ and applied in materials science to find descriptors of complex material properties.^{39,40} It has also previously been used to identify simple interatomic potentials.^{9,41–43} Here, we go beyond these previous efforts by demonstrating that genetic programming is capable of finding fast, accurate and transferable many-body potentials for a metallic system from ab-initio calculations.

The key to our approach is the construction of a physically meaningful hypothesis space, achieved by analyzing interatomic potentials that were derived from physical principles.^{25,27,28} We take advantage of natural similarities in the functional forms of simple, physics-derived models²⁵ to construct a hypothesis space that contains many such functional forms. The hypothesis space that we use consists of all functions that can be constructed from combinations of addition, subtraction, multiplication, division, and power operators; constant values and distances between atoms; and an operator that performs a sum over functions of distances

¹Department of Materials Science and Engineering, Johns Hopkins University, Baltimore, MD, USA. *email: tmueller@jhu.edu

between a given atom and all neighbors within a given cutoff radius. This space contains a wide variety of potential models derived from fundamental physical interactions, including nearly all pair potentials (e.g., Lennard-Jones,⁴⁴ Coulomb,⁴⁵ Morse⁴⁶), as well as many-body glue potentials,²⁵ bond-order potentials (without the bond angle terms),^{25,26,47,48} and combinations thereof. Even for relatively simple hypothesis spaces such as this one, it is difficult to enumerate a list of even relatively simple functional forms that can be created due to the large number of ways in which the various operators and values can be combined.⁴⁰ Here, we use a genetic algorithm and multi-objective optimization to search this hypothesis space for interatomic potentials that are simple (and thus more likely to be generalizable⁴⁹), fast, and accurate. Additional details of our approach are provided in the Methods section.

RESULTS

Validating the machine-learning algorithm

To validate our algorithm, we tested its ability to rediscover the exact form of two interatomic potentials: the Lennard-Jones potential and the Sutton-Chen (SC) EAM potential. In each case, the genetic algorithm was able to identify the exact function used to generate the training data. The training data for the Lennard-Jones potential were generated by taking 75 snapshots (1 snapshot every 5000 steps with a time step of 1 fs) of 32-atom molecular dynamics simulations: 15 snapshots at 80 K in a canonical (NVT) ensemble, 15 snapshots at 80 K and 100 kPa in an isothermal-isobaric (NPT) ensemble, 15 snapshots at 100 K (NVT), 15 snapshots at 100 K and 100 kPa (NPT) and 15 snapshots at 20,000 K (NVT). It consisted of 75 energies and 7200 components of force,⁵⁰ generated using the following parameterized model for argon:⁵¹

$$V_{\text{LJ}} = \sum_i \sum_j \left(\frac{49304.15}{r^{12}} - \frac{34.88}{r^6} \right) \quad (1)$$

where V_{LJ} is the potential energy of the system, the index i represents an atom in the structure, j is its neighbor and r is the distance between the two atoms. The genetic programming algorithm found:

$$V = \sum_i \left(-50.18(983.04) \left(\sum_j (3.35r)^{-6.00} - \sum_j r^{-12.00} \right) \right) \quad (2)$$

which simplifies to the form of the Lennard-Jones potential in Eq. (1).

The training data for the SC EAM potential were obtained from 100 snapshots (1 snapshot every 100 steps with a time step of 1 fs) of 32-atom molecular dynamics simulations: 25 snapshots at 300 K, 25 snapshots at 1600 K, 25 snapshots at 3800 K, and 25 snapshots at 20,000 K, all in the NVT ensemble. The training set consisted of 100 energies and 9600 components of force. The potential used to generate the training data was parametrized for copper:

$$V_{\text{SC}} = \sum_i \left(\sum_j \frac{644.52}{r^9} - \left(\sum_j \frac{527.62}{r^6} \right)^{0.5} \right) \quad (3)$$

The artificial intelligence algorithm found:

$$V = \sum_i \left(-0.73 - 2.53 \left(\left(-0.66(384.39) \sum_j r^{-9.00} \right) + \left(0.25 / \left(20.63 \sum_j r^{-6.00} \right) \right)^{-0.50} \right) \right) \quad (4)$$

When it is simplified, it gives the same form as V_{SC} with a constant shift and a slight difference between the constant parameters that could be eliminated by tightening the convergence criterion for parameter optimization. The values of the

parameters in the exponents were found to the second decimal place.

Discovering new models for copper

Having established that our genetic programming algorithm can find the exact form of simple pair and many-body potentials, we evaluated its ability to find potential models from data generated using density functional theory⁵² (DFT). For this purpose, we generated 150 snapshots (one snapshot every 100 steps with a time step of 1 fs) of 32-atom DFT molecular dynamics simulations on fcc copper: 50 snapshots at 300 K (NVT), 50 snapshots at 1400 K (NVT) and 50 snapshots at 1400 K (NPT at 100 kPa). The copper had melted and lost its fcc structure for the simulations at 1400 K. The data consisted of 150 energies, 14,400 components of forces and 900 components of virial stress tensors.⁵³ One half was randomly selected for training and the other half for validation. Models were evaluated on three metrics: complexity, defined as the number of nodes on the model; computational cost, defined as the number of summations over neighbors, as these typically consume most of the execution time; and fitness, defined as a weighted sum of the mean squared errors of the energies, forces and stresses, which were normalized to unitless values as described in the methods section:

$$\text{Fitness} = 1000 * (0.5\text{MSE}_{\text{energy}} + 0.4\text{MSE}_{\text{force}} + 0.1\text{MSE}_{\text{stress}}) \quad (5)$$

To identify promising models we constructed a three-parameter convex hull based on fitness, computational cost, and complexity. Some of the models on this hull are shown in Table 1.

Many of the models discovered by the genetic programming algorithm have forms that resemble the embedded-atom model, or “glue” type potentials. The models consist of a sum of a pairwise term with a repulsive component and a many-body “glue” type attractive term, which consists of a nonlinear transformation (an “embedding” function) of a sum over neighbors (the “density”). Here, we select two of the models, which we label GP1 and GP2, for further analysis based on their favorable tradeoff between simplicity and their prediction errors for the elastic constants (Supplementary Table S9). In GP1, the simpler of the two models, the embedding function is simply the inverse of the density. In GP2, the embedding function is the same, and it is multiplied by a sum of pairwise interactions to form the glue term. Although GP1 and GP2 resemble known potential models, there are some notable differences. They have much simpler functional forms than most other copper potential models, and they have a different form for the attractive “glue” part of the potential. It is common in EAM-type potential models for the embedding function to be the negative square root of the density; this can be derived from the second moment approximation.²⁵ In GP1 and GP2, the attractive term instead depends on the positive inverse of a sum over pairwise interactions. Unlike the other models, this embedding function is bounded in the limit of high densities and diverges to infinity in the limit of zero density. GP1 and GP2 also include terms with the unusual form of r^{a-br} , which grows by a power law before decaying superexponentially. The resulting models demonstrate high-predictive power for condensed phases and defects that were not included in the training data and, even though there were no surfaces in the data used to train them, they largely avoid the severe underprediction of surface energies that is common for embedded-atom type models (Supplementary Table S6).⁵⁴

Validating and evaluating the transferability of the interatomic potentials. As might be expected by their simplicity, neither GP1 nor GP2 overfit their training data. For each model, there is little difference between the training mean absolute error and validation mean absolute error for energies, components of force vectors and components of the virial stress tensors (Fig. 1). As an

Table 1. The three-dimensional convex hull of models found by the machine-learning algorithm

Fitness	Cost ^a	Complexity	Expression
5393157	1	2	$\sum r f(r)$
1800.1	1	4	$\sum r^{-3.20} f(r)$
105.30	1	8	$\sum (649.17r^{-9.83} - 0.09)f(r)$
54.144	1	10	$\sum (r^{10.20-5.49r} - 0.07)f(r)$
26.906	2	13	$\sum r^{10.20-5.49r} f(r) + 33.77(\sum f(r))^{-1}$
8.1584	2	15	$\sum r^{10.21-5.48r} f(r) + 1.19(\sum 0.33'f(r))^{-1}$
7.8230 ^b	2	21	$\sum (r^{10.21-5.47r} - 0.21')f(r) + 0.97(\sum 0.33'f(r))^{-1}$
7.8229	2	25	$0.999\sum (r^{10.21-5.46r} - 0.21')f(r) + 0.97(\sum 0.33'f(r))^{-1} + 5.76$
7.4131	4	19	$\sum r^{10.21-5.48r} f(r) + (3.07\sum f(r))(\sum 0.31'f(r))^{-1}(\sum r f(r))^{-1}$
4.7294 ^b	3	28	$7.33\sum r^{3.98-3.94r} f(r) + (27.32 - \sum (11.13 + 0.03r^{11.74-2.93r})f(r))(\sum f(r))^{-1}$
4.2932	4	29	$6.76\sum r^{4.00-3.88r} f(r) + 17.25(\sum f(r))(\sum r^{11.68-3.07r} f(r))^{-1} + 25.30(\sum f(r))^{-1}$

$f(r)$ is the smoothing function defined in Eq. (7). ^aCost is based on the number of summations over neighbors. ^bThe models with fitness 7.8230 and 4.7294 are named GP1 and GP2, respectively

initial comparison of the performance between GP1, GP2, and other similar potential models, we evaluate how well they predict the elastic constants of fcc copper. The elastic constants C11, C22, and C44 are a widely used benchmark of copper potential model performance, allowing us to make a comparison between nine different copper potential models for which elastic constant data is available. We have plotted the maximum percent error in predicted elastic constants against the complexity of the model, as measured by number of nodes, in Fig. 2. These errors, and all errors listed in this paper, are measured against each model's own target values, which are provided in the Supplementary Information. The potentials discovered by the machine-learning algorithm presented in this work significantly change the Pareto frontier of interatomic potentials, defined as the set of interatomic potentials for which no other potential has less error and is less complex. They have errors comparable to the most accurate potential models and complexity comparable to the simplest (Table 2).

Both GP1 and GP2 reproduce the radial distribution function of a liquid state well (Fig. 3), which is likely partially due to the inclusion of snapshots of the liquid state in their training data. There is also good agreement between the newly discovered potential models and other DFT-calculated properties (Table 3). Other models near the Pareto frontiers also show good agreement with their target values, but a notable difference is other than being more complex, these models were also directly trained on many of the properties listed in Table 3 whereas GP1 and GP2 were not. The errors on the elastic constants predicted by GP2 are almost as small as for EAM1, and the simpler model GP1 has errors on elastic constants that are comparable to ABCHM. The GP1 and GP2 models perform well on properties involving hcp and bcc phases, even though no hcp or bcc data were included in the training set. For the bcc lattice constant, the relative energy between the fcc and bcc phases, and the relative energy between fcc and hcp phases, GP1 and GP2 perform comparably to models that were trained on those data points and outperform all models that were not trained on them.

For vacancy formation energies in the dilute limit, GP2 performs very well, with an error of 2 meV relative to the extrapolated DFT energy (see Supplementary Information for details). GP1 performs less well, with an error of 138 meV. Comparisons with other models for vacancy formation energies are difficult, as the models that report their performance on vacancy formation energies were trained with those values, whereas GP1 and GP2 were not. An exception is a neural network potential we discuss later, for which the extrapolated error is 146 meV, comparable to GP1 (Table S6, Supplementary Information). The GP1 error in vacancy formation

energy is largely offset by an error in the opposite direction for migration energy, and as a result the errors for both GP1 and GP2 for the activation energy for vacancy-mediated diffusion are comparable to models that were trained on that value.

GP1 and GP2 also demonstrate good predictive accuracy on phonon frequencies, which were also not included in their training set (Fig. 4 and Table 3). On average, GP2 outperforms all other models on phonon frequencies used for testing, with a mean absolute error of 2.0%, and it also outperforms EAM1 on the phonon frequencies on which EAM1 was trained. GP1 does not do as well as GP2 on phonon frequencies, performing on average slightly better than EAM2 but worse than EAM1 and CuNi. The difference in the performance of GP1 and GP2 on phonons is evident in their calculated phonon dispersion curves (Fig. 4). The strong performance of GP2 on phonon frequencies and elastic constants suggests that it does well at capturing the curvature of local minima on the potential energy surface, but it may not do as well in states away from the local minima, such as the vacancy formation energy of an unrelaxed $2 \times 2 \times 2$ fcc unit cell (Table 3).

Both GP1 and GP2 perform better than the other models for the formation energy of a dumbbell defect. The absolute errors for GP1 and GP2 are only 49 meV and 56 meV, respectively, as compared to an absolute prediction error of 250 meV for the ABCHM model (Table 3). Of the three models that included the dumbbell defect formation energy in their training data, the best has an absolute error that is about twice that of the absolute prediction error of GP2 (Supplementary Table S4). On the other hand, both GP1 and GP2 underestimate the formation energy of a stable intrinsic stacking fault (see Supplementary Information for details) to a greater extent than the other models that report a comparison to this value. The largest absolute error, 29 mJ/m² for GP1, is 10.2 meV/atom along the (111) plane of the fault. GP1 and GP2 similarly underestimate the formation energy of an unstable stacking fault, but it is hard to assess how this compares to other models as none of the other models reports a benchmark value for the unstable stacking fault energy.

EAM-type models are well known to underpredict surface energies. Surface energies predicted by EAM-type models trained on ab-initio calculations for copper are about 40–50% below their target values for the (100), (110) and (111) surfaces (Supplementary Table S6). In contrast, GP1 underpredicts these surface energies by only 8%, 1%, and 5% respectively, and GP2 underpredicts them by 14%, 10%, and 10%, respectively (Fig. 5). For potentials that use experimental data for their target values, evaluating performance in calculating surface energies is more difficult as only the average value of experimental surface energies is available.^{54–56} To make

Fig. 1 Parity plots of training (orange) and validation (blue) energies, components of force and components of the virial stress tensor for the interatomic potential GP1 (first row) and GP2 (second row). The black dashed line is the identity. The mean absolute error (MAE) is presented above each sub-figure for validation and training data, respectively

Fig. 2 Pareto frontiers of interatomic potentials for copper. No model has less error and is less complex than a model in the Pareto frontier. The orange dashed line was the Pareto frontier before the development of GP1 and GP2, and the blue dashed line is the new Pareto frontier. The percent error for each model was evaluated against the model's own target values, described in the Supplementary Information. Complexity was measured by the number of nodes in the tree representation of the model. As the smoothing function for some models is unknown, to construct this plot each smoothing function was counted as 2 nodes, representing the smoothing function and a multiplication operation. Sources: SC,⁷⁹ ABCHM,⁵⁸ Cu1,⁵⁸ EAM1,⁵⁴ EAM2,⁵⁴ Cu2,⁸⁰ Cuu6,⁸¹ Cuu3⁵⁶ and CuNi.⁵⁵ The interatomic potentials were found in the Interatomic Potentials Repository.⁸² GP1 and GP2 were developed in this work

this comparison we have calculated weighted average surface energies over 13 different low-index surface facets, where the weights are based on the relative surface areas in Wulff constructions (details are provided in the Supplementary Information).⁵⁷ EAM1 and Cuu3 underpredict the weighted surface energies by about 30%, and CuNi overpredicts the weighted surface energies by about 10% (Supplementary Table S6).⁵⁸ GP1 underpredicts the weighted surface energies by 8% and GP2 by 13%. GP1-predicted surface energies are the most accurate of any of the evaluated EAM-type potential models relative to its target values.

The performance of GP1 and GP2 on surface energies is remarkable because there were no surfaces in the training set; this is a case of machine-learning potential models demonstrating extrapolative predictive ability. Similarly, both GP1 and GP2 demonstrated high-predictive accuracy for the dumbbell defect compared to the other models, indicating that they are able to accurately predict energies in both low-coordination and high-coordination environments. There are likely two reasons for the predictive accuracy of these models. The first is that other than SC, GP1, and GP2 are the simplest models considered here, and in general simpler models are less likely to overfit the training data.⁴⁹ A similar trend of simpler models demonstrating greater extrapolative ability was observed by Yunxing et al.¹⁹ in a recent comparison of different types of machine learned potential models. The second reason is that these models were discovered in a hypothesis space designed to contain models resembling those for which there is fundamental physical justification. In general, the more physics can be included in the machine-learning procedure, the more likely it is that a model will have extrapolative predictive power.

When new data are added to the training set, the genetic programming search for new models can build off of what has

Table 2. Interatomic potentials near the Pareto frontiers in Fig. 2

Name	Expression
SC ⁷⁹	$E_i = \sum_j \frac{644.52}{r^3} f(r) - \left(\sum_j \frac{527.62}{r^6} f(r) \right)^{0.5}$
GP1	$E_i = \sum_j (r^{10.21-5.47r} - 0.21^r) f(r) + 0.97 \left(\sum_j 0.33^r f(r) \right)^{-1}$
GP2	$E_i = 7.33 \sum_j r^{3.98-3.94r} f(r) + \left(27.32 - \sum_j (11.13 + 0.03r^{11.74-2.93r}) f(r) \right) \left(\sum_j f(r) \right)^{-1}$
EAM2 ⁵⁴	$E_i = \sum_j E_1 (e^{-2a(r-r_0)} - 2e^{-a(r-r_0)}) f(r) + F \left(\sum_j r^6 (e^{-\beta r} + 2^9 e^{-2\beta r}) f(r) \right)$ $\sum_i F(\bar{\rho}_i) = E(L) - \frac{1}{2} \sum_i \sum_j E_1 (e^{-2a(r-r_0)} - 2e^{-a(r-r_0)}) f(r)$ $E(L) = -E_{\text{sub}}(1 + a^*)e^{-a^*}$ $a^* = (a/a_0 - 1)/(E_{\text{sub}}/9B\Omega)^{1/2}$
ABCHM ⁵⁸	$E_i = \sum_j \varphi(r) f(r) + 1.57 \cdot 10^{-5} \left(\sum_j \psi(r) f(r) \right)^2 - \left(\sum_j \psi(r) f(r) \right)^{0.5}$ $\varphi(r) = \begin{cases} e^{0.82+16.01r-15.73r^2+3.80r^3}, & 1 < r < 1.9 \\ +0.62(4.43-r)^3, & 1.9 < r < 4.43 \\ -3.02(4.17-r)^3, & 1.9 < r < 4.17 \\ +2.84(4.04-r)^3, & 1.9 < r < 4.04 \\ -0.41(3.62-r)^3, & 1.9 < r < 3.62 \\ +0.65(3.13-r)^3, & 1.9 < r < 3.13 \\ +0.81(2.56-r)^3, & 1.9 < r < 2.56 \end{cases}$ $\psi(r) = \begin{cases} 0.21(4.43-r)^3, & 1.9 < r < 4.43 \\ +0.36(3.62-r)^3, & 1.9 < r < 3.62 \end{cases}$
CuNi ⁵⁵	$E_i = \frac{1}{2} \sum_j (D_M [1 - e^{-a_M(r-R_M)}]^2 - D_M) f(r) + F(\bar{\rho}_i)$ $\bar{\rho}_i = \sum_j \tanh(20r^2) \left\{ r^6 (e^{-\beta r} + 2^9 e^{-2\beta r}) + \frac{\sigma^{(1)}}{\mu^{(1)}} e^{-\frac{1}{2}[\mu^{(1)}(r-R_\theta)]^2} - 0.1\sigma^{(1)} e^{-\frac{1}{2}[\mu^{(1)}(r-(R_\theta+0.5))]} \right\} f(r)$ $\sum_i F(\bar{\rho}_i) = E(L) - \frac{1}{2} \sum_i \sum_j (D_M [1 - e^{-a_M(r-R_M)}]^2 - D_M) f(r)$ $E(L) = -E_{\text{sub}}(1 + a^*)e^{-a^*}$ $a^* = (a/a_0 - 1)/(E_{\text{sub}}/9B\Omega)^{1/2}$
EAM1 ⁵⁴	$E_i = \sum_j \left(\left[\frac{E_1 (e^{-2a_1(r-r_0^{(1)})} - 2e^{-a_1(r-r_0^{(1)})})}{E_2 (e^{-2a_2(r-r_0^{(2)})} - 2e^{-a_2(r-r_0^{(2)})})} + \delta \right] f(r) - \sum_{n=1}^3 (H(r_s^{(n)} - r) S_n(r_s^{(n)} - r)^4) \right) + F(\bar{\rho}_i)$ $\text{if } (\bar{\rho}_i < 1) : F(\bar{\rho}_i) = F^{(0)} + 0.5F^{(2)}(\bar{\rho}_i - 1)^2 + \sum_{n=1}^4 (q_n(\bar{\rho}_i - 1)^{n+2})$ $\text{else} : F(\bar{\rho}_i) = \frac{F^{(0)} + 0.5F^{(2)}(\bar{\rho}_i - 1)^2 + q_1(\bar{\rho}_i - 1)^3 + Q_1(\bar{\rho}_i - 1)^4}{1 + Q_2(\bar{\rho}_i - 1)^3}$ $\text{where} : \bar{\rho}_i = \sum_j \left([ae^{-\beta_1(r-r_0^{(3)})^2} + e^{-\beta_2(r-r_0^{(4)})}] f(r) \right)$

All potentials are in units of eV and Å. $f(r)$ is a smoothing function; for GP1 and GP1 it is defined in Eq. (7). EAM2 and CuNi defined the embedding function to match a universal equation of state⁸³

been previously learned by using known high-performing models to seed the search. As a demonstration of this approach, we have performed an additional search using an augmented training set in which the 13 low-index surfaces (shown in Fig. 5) were added to the training data and always included in the subsets of data used to evaluate candidate models. This search was seeded with GP1 and GP2, and as a result the models it discovered (Supplementary Table S10) had many features in common with these. One of these models, which we label GP3 (Eq. (6)) resembles GP2 but, as expected, demonstrates better performance on surface energies (Fig. 5). The absolute error for the weighted surface energies is 7% for GP3, as compared to 13% for GP2. The equation for GP3, which

is slightly simpler than that of GP2, is provided below.

$$7.51 \sum_j r^{3.98-3.93r} f(r) + \left(28.01 - 0.03 \sum_j r^{11.73-2.93r} f(r) \right) \left(\sum_j f(r) \right)^{-1} \quad (6)$$

On average, the improved performance on surface energies for GP3 does not significantly affect its performance on the other properties listed in Table 3 compared to GP2. GP3 performs worse on average on elastic constants and phonon frequencies, but significantly better on the dumbbell formation energy and stacking fault energies. It is difficult to assess the extent to which

Fig. 3 Radial distribution functions of liquid copper at 1400 K

Fig. 4 Calculated phonon dispersion curves for DFT, GP1, GP2, and GP3

Table 3. Error of the values predicted by interatomic potentials for copper relative to the respective reference

Property	Metric	SC	GP1	GP3	GP2	EAM2	ABCHM	CuNi	EAM1
Complexity	Number of nodes	15	21	26	28	113	146	150	158
C_{11}	% Error	-3.6^a	5.8^b	2.9^b	-0.7^b	1.9	-0.6	0.1	-0.1
C_{12}	% Error	3.8^a	7.0^b	2.5^b	0.5^b	0.2	-4.1	0.0	0.1
C_{44}	% Error	-29.4	-2.0^b	-0.4^b	-1.2^b	0.0	-6.6	0.0	0.5
a_0 (fcc)	% Error	0.0	-0.3	0.2	0.3	0.0	-0.7	0.0	0.0
a_0 (bcc)	% Error	-	0.1	-0.2	-0.1	-	2.4	0.9	-
ΔE (bcc - fcc)	Pred. - ref. (meV/atom)	-	8	12	4	2^c	-11	-13	2
ΔE (hcp - fcc)	Pred. - ref. (meV/atom)	-	-3	-1	-2	-6^c	-2	-4	-4
$E_{v(\text{unrelaxed}, 2 \times 2 \times 2)}$	Pred. - ref. (meV)	-	32	-106	-123	-	80	-	-
E_v	Pred. - ref. (meV)	-	138	12	2	-17	-	6	-3
E_m	Pred. - ref. (meV)	-	-106	-49	-37	-20	-	-20	-21
$E_a = E_v + E_m$	Pred. - ref. (meV)	-	32	-36	-34	-37	-24^d	-14^d	-24
E_{dumbbell}	Pred. - ref. (meV)	-	49	-15	-56	-	250	-	-
$v_L(X)$	% Error	-	8.2	4.1	3.2	7.6	-	0.0	6.0
$v_T(X)$	% Error	-	0.7	0.1	0.0	1.2	-	-0.2	0.8
$v_L(L)$	% Error	-	6.5	1.7	0.5	6.6	-	-0.8	4.6
$v_T(L)$	% Error	-	-2.2	-2.3	-3.0	-1.5	-	-9.4	-2.6
$v_L(K)$	% Error	-	9.1	5.1	4.3	8.0	-	-0.8	5.4
$v_{T1}(K)$	% Error	-	1.6	0.3	0.1	2.4	-	1.1	1.1
$v_{T2}(K)$	% Error	-	6.9	3.0	2.2	9.0	-	0.9	7.0
γ_{ISF}	Pred. - ref. (mJ/m ²)	-	-29	-6	-20	-9	-	0	-1
γ_{USF}	Pred. - ref. (mJ/m ²)	-	-44	-27	-31	-	-	-	-

The models displayed in this table are near the Pareto frontiers in Fig. 2, values of other potentials are in Supplementary Tables S2 to S7. C_{ij} are elastic constants, a_0 is the lattice parameter, ΔE (bcc - fcc) is the energy difference between bcc and fcc phases, E_v is the fcc bulk vacancy formation energy, $E_{v(\text{unrelaxed}, 2 \times 2 \times 2)}$ is the unrelaxed vacancy formation energy computed on a $2 \times 2 \times 2$ supercell, E_m is the migration energy for fcc bulk vacancy diffusion, E_a is the activation energy for fcc bulk vacancy diffusion, E_{dumbbell} is the dumbbell <100> formation energy, ν is the phonon frequency, and γ_{ISF} and γ_{USF} are the intrinsic and unstable stacking fault energies, respectively.

Properties in *italics* font were used for training, and properties in **bold** font were not used for training. Properties for which target values are not available are marked with a “-”.

^aSC was fit to the bulk modulus

^bElastic constants were used to select GP1, GP2, and GP3 from the convex hull

^cFit to ensure that $E_{\text{fcc}} < E_{\text{bcc}}$ and $E_{\text{fcc}} < E_{\text{hcp}}$

^dFit to vacancy formation energy

Fig. 5 Surface energies of elemental copper as computed using DFT, and the interatomic potentials GP1, GP2, and GP3

these changes in performance can be attributed to the addition of surfaces to the training data due to the stochastic nature of the search.

Although GP1, GP2, and GP3 are simpler than many other EAM-type models, they have a similar computational cost when implemented in LAMMPS^{29,59} due to the extensive use of tabulated values. Based on our benchmarks (Supplementary Fig. S1) GP1 takes 2.1 $\mu\text{s}/\text{step}/\text{atom}$, GP2 3.5 $\mu\text{s}/\text{step}/\text{atom}$, and GP3 takes 3.6 $\mu\text{s}/\text{step}/\text{atom}$, whereas EAM1 has a cost of 3.0 $\mu\text{s}/\text{step}/\text{atom}$. These speeds rank them among the fastest potential models, capable of modeling systems at large time and length scales.²⁹

DISCUSSION

There are advantages and disadvantages to the different approaches for using machine learning to generate potential models. In many machine-learning approaches, including (but not limited to) neural network potentials, Gaussian approximation potentials, moment tensor potentials, SNAP potentials, and AGNI force fields,^{1–5} the general idea is to construct a highly flexible hypothesis space that respects local symmetry and, with the help of large amounts of training data, identify the models within that hypothesis space that best reproduce the training data. Such models are capable of achieving very high-accuracy for systems in which the local environments of the atoms are similar to those contained in the training set. These machine-learning algorithms typically produce potential models that are orders of magnitude faster than DFT but also orders of magnitude slower than EAM-type potentials.^{29,53,60–62}

Here, we have demonstrated that machine learning can also be used to develop the types of simple, fast potential models that are needed to model systems at extreme time and length scales. The key to our approach is to use genetic programming to search for computationally simple and efficient models in a hypothesis space that is constructed so that it contains simple models that are also physically meaningful. The models are then selected based on a combination of simplicity, speed, and accuracy relative to the training data. The use of simplicity as a selection criterion results in models that are more likely to generalize well, and it also significantly reduces the amount of data required to train the model.^{63–65} For example, GP1 and GP2 were trained with 75 32-atom structures, for a total of 2400 atomic environments. For comparison, Artrith and Behler⁶⁶ have constructed a neural

network potential for copper with a focus on surfaces. The potential was trained using 554,187 atomic environments, including tens of thousands of slabs and cluster structures. It performs comparably to GP1 and GP2 for many bulk properties, and much better for surface energies (Table S6, Supplementary Information). The neural network approach demonstrates very low errors on the types of systems on which it was trained, but as the genetic programming approach requires less training data it is likely that some accuracy can be recovered by using more accurate (and computationally expensive) methods to generate the training data.

The potential models discovered by the genetic programming approach are as fast as EAM-type models and demonstrate good predictive accuracy on properties they were not trained on. In particular, GP1 and GP2 show surprisingly good performance in predicting surface energies (the GP1 mean absolute error for surface energies is only 35 mJ/m^2) despite the fact that there were no surfaces in their training data. Trained only on DFT data, the genetic programming algorithm found models that resemble widely used glue potentials with a unique form for the many-body term that depends on the inverse of a sum over pair interactions. One of the advantages of generating potential models using simple analytical expressions is that it may be possible to analyze the expressions to get an insight into the underlying physical interactions that are responsible for the shape of the potential energy surface.

There are some notable limitations and areas for improvement for the approach presented here. For each system studied, it will be necessary to ensure that the hypothesis space contains simple expressions that capture important contributions to the potential energy; for example, for many systems it will likely be necessary to introduce terms that depend on bond angles, which was not done in this work. We used fixed inner and outer cutoff distances in this study, but it would almost certainly be better to let them vary as do other parameters of the potential. There is also the question of how to determine which of the models discovered by the genetic programming algorithm provide the best balance of speed and predictive accuracy. This could be achieved in a number of ways,^{40,67,68} including by evaluating performance against validation data, but it is not clear which approach is best. Finally, the genetic programming approach is likely not suitable for on-the-fly learning. As it is a stochastic method, it can take an indeterminate amount of time to find a set of promising models, and there is no guarantee that an incremental change to the training data will result in an incremental change to the shapes of the potential energy surfaces on the convex hull. Other potential model approaches are probably better-suited for this purpose. Despite these current limitations, our results demonstrate that machine learning holds great promise to improve the accuracy of atomistic calculations at extreme time and length scales.

METHODS

Description of the hypothesis space

Our machine-learning algorithm uses genetic programming to search a hypothesis space of models that can be constructed by combining real numbers, addition, subtraction, multiplication, division, exponentiation, and a sum over neighbors of an atom. As discussed previously in the text, the hypothesis space was based on physical principles. Within this hypothesis space, each function can be represented as a tree graph, as shown in Fig. 6. The space was constrained so that the maximum number of summations over neighbors was 6, no nested summations over neighbors were allowed, the maximum allowed depth of a tree was 32 and the maximum allowed number of nodes was 511. To ensure smoothness of the potential, all functions of distances are multiplied by the following smoothing function before the sum over neighbors is taken:⁶⁹

$$f(r) = (2r^2 - 3r_{\text{in}}^2 + r_{\text{out}}^2)(r_{\text{out}}^2 - r^2)^2(r_{\text{out}}^2 - r_{\text{in}}^2)^{-3} \quad (7)$$

Fig. 6 Tree graphs of **a**) a Lennard-Jones potential parametrized for argon, Eq. (1), **b**) a Sutton-Chen EAM potential parametrized for copper, Eq. (3), **c**) GP1 and **d**) GP2

where r_{in} and r_{out} are the inner and outer cutoff radii, for GP1 and GP2, $r_{in} = 3 \text{ \AA}$ and $r_{out} = 5 \text{ \AA}$, including the 3rd nearest neighbors.⁵⁴

Description of the algorithm

Genetic programming evolves computer programs following Darwin's natural selection by performing crossover and mutation operations on a set of individuals. Crossover was performed by two different operations: by randomly selecting a branch from one tree and replacing it with a randomly selected branch of another tree (Fig. 7), and by creating a linear combination of two randomly selected branches from two different trees—the first method was randomly selected 90% of the time and the second one 10% of the time. The mutation operation performed three different sub-operations with equal probability: crossover of a tree with a randomly generated tree, swapping the arguments of a binary non-commutative function, and slightly modifying the expression tree by replacing (or inserting) a randomly selected non-terminal node with a randomly selected operator.⁷⁰ The randomly generated trees were generated with the grow or full method with equal probability,³⁶ and the depth was drawn from a Gaussian distribution of mean 5 and standard deviation of 1. The overall algorithm performed crossover with a probability of 0.9, and mutation with a probability of 0.1.

Increasing diversity is known to improve the quality of the optimization.⁷⁰ To increase diversity, we implemented a hierarchical way of creating separate environments in which the individuals (i.e., potential models) evolved. We ran the algorithm on 12 processors, and each processor had its own environment, consisting of a population of models and a subset of the training data. Conceptually this allows potentials within a specific

environment to develop characteristics that are unique, increasing the diversity. Candidates for crossover and mutation were selected from 3 different sets of models with equal probability:

- (1) The population of the current processor. Every 20,000 crossover and mutation operations, 100 individuals were selected based on their fitness (Eq. (5)) with Pareto tournament selection of size 10 while the rest were discarded.^{67,71,72}
- (2) A global set of models. Each processor tried to add the 100 individuals selected in part (1) to the global subset every 20,000 crossover and mutation operations. The models on the global set were then evaluated on the basis of speed (to model large time and length scales), fitness (for accurate results), and complexity (for generalizability). The speed of each model was estimated by the number of summations over neighbors. The complexity was evaluated by the number of nodes in the tree graph. To identify the best models, we generated separate convex hulls with respect to fitness and complexity for each number of summations (speed) in a potential. Only the models on these convex hulls were retained in the global set.
- (3) Individuals from other processors. Each processor was allowed to communicate with other processors every 5000 crossover and mutation operations, importing the current set of individuals from them.

Selection with equal probability was performed when getting an individual from the global set. Tournament selection of size 10 was used for getting individuals from the population of the current processor and from the populations of other processors.

Fig. 7 Example of a crossover operation

The training data was also arranged in hierarchical subsets to increase diversity and reduce the speed of evaluating fitness. Globally, a subset of 75 energies, 75 forces, and 75 stresses was randomly sampled from the full set of training data every 20,000 crossover and mutation operations. The fitness of the global set of models was evaluated using this subset of training data. The training data on each processor (15–30 energies, forces and stresses) were randomly selected from the global subset of the training data, and this local subset was used to evaluate fitness locally on each processor. The subset of training data for each processor was selected from the global subset because individuals that migrate from a processor to the global set are more likely to survive if the environment is similar.

Optimization of potential model parameters was performed using the covariance matrix adaptation evolution strategy (CMA-ES) optimizer and a conjugate gradient (CG) optimizer.^{73,74} The CMA-ES algorithm was selected because it performs well in nonlinear or non-convex problems. The potential models on the global set of best individuals were optimized with the CMA-ES every 10,000 crossover and mutation operations by one processor. In contrast, the CG algorithm performed one optimization step for every individual generated by crossover or mutation.

The genetic programming algorithm took about 330 CPU-hours to find the exact Lennard-Jones potential, 3600 CPU-hours to find the exact Sutton-Chen potential, and 360 CPU-hours to find GP1, GP2, and GP3. We note that it is likely that with additional tuning and performance enhancements the efficiency of the algorithm can be improved. To facilitate this, our code is open-source and available at <https://gitlab.com/muellergroup/poet>.

Details about the target data

The DFT data were computed using the Vienna Ab-initio Simulation Package⁷⁵ (VASP) with the Perdew–Burke–Ernzerhof⁷⁶ (PBE) generalized gradient approximation (GGA) exchange correlation functional. The projector augmented wave method⁷⁷ (PAW) Cu_pv pseudopotential was used for copper. Efficient *k*-point grids were obtained from the *k*-point grid server with MINDISTANCE = 50 Å.⁷⁸ A cutoff energy of 750 eV and ADDGRID = TRUE in VASP were required to converge the stress tensor to less than 0.05 GPa. The elastic constants were converged to within 3 [% error] using a MINDISTANCE = 100 Å. The DFT point defect energies were computed by linear extrapolation (see Supplementary Information for more details). The phonon dispersion curves were computed on a 3 × 3 × 3 supercell. The DFT calculation used a 5 × 5 × 5 *k*-point grid and electronic self-consistency convergence of 10^{−8} eV. The radial distribution function molecular dynamics simulations were performed in the NVT ensemble at the experimental 1400 K liquid density on a 3 × 3 × 3 supercell. The temperature was increased from 300 K to 2500 K during 1 ps. Then the temperature was maintained at 2500 K during 10 ps. Then, the temperature was decreased from 2500 K to 1400 K over 1 ps. Then the temperature was maintained at 1400 K during 1 ps. Finally, the radial distribution function data was collected at 1400 K over 40 ps. The DFT molecular dynamics for the radial distribution function was performed with a cutoff energy of 400 eV for the equilibration steps and 750 eV for the final 40 ps during which data was collected. Electronic self-consistency convergence was 10^{−5} eV and only the *k*-point at Γ was used. For the computation of the fitness of the models, the energies were transformed by subtracting the minimum and dividing by the standard deviation, and the forces and stresses were standardized by subtracting the mean and dividing by the standard deviation.

The data used to rediscover the Lennard-Jones potential and the SC potential, and the data used to validate GP1 and GP2 were computed with LAMMPS. Instructions and files required to use GP1 and GP2 with LAMMPS

are provided on the Supplementary Information. Lennard-Jones calculations used a cutoff distance of 7.5 Å, and SC, GP1, GP2, and GP3 calculations used a cutoff distance of 5 Å.

Benchmarking model speed

Benchmarking of model speed was done on a single core of a Haswell node with a clock speed of 2.5 GHz. The benchmarking simulation consisted of 10,000,000 molecular dynamics steps for a 32-atom unit cell.

DATA AVAILABILITY

The data used to train and test the models is available along with the open-source code at https://gitlab.com/muellergroup/poet/tree/master/poet_run/data. The Supplementary Information includes: tables of the errors of the predictions of the interatomic potentials on elastic constants, lattice parameters, energy difference between phases, vacancy formation energies, migration and activation energies, and surface energies; a figure (Supplementary Fig. S1) showing the computational cost of LJ, SC, GP1, GP2, GP3, and EAM1 as implemented in LAMMPS; a figure (Supplementary Fig. S2) showing that the Pareto frontier of absolute percent error on elastic constants against complexity does not change when using the average instead of the maximum error; the instructions and the files required for enabling GP1, GP2 and GP3 in LAMMPS.

CODE AVAILABILITY

Our code is open-source and available at <https://gitlab.com/muellergroup/poet>.

Received: 18 March 2019; Accepted: 25 October 2019;
Published online: 18 November 2019

REFERENCES

- Bartók, A. P., Payne, M. C., Kondor, R. & Csányi, G. Gaussian approximation potentials: the accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.* **104**, 136403 (2010).
- Thompson, A. P., Swiler, L. P., Trott, C. R., Foiles, S. M. & Tucker, G. J. Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials. *J. Comput. Phys.* **285**, 316–330 (2015).
- Behler, J. & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **98**, 146401 (2007).
- Huan, T. D. et al. A universal strategy for the creation of machine learning-based atomistic force fields. *NPJ Computational Mater.* **3**, 37 (2017).
- Shapeev, A. Moment tensor potentials: a class of systematically improvable interatomic potentials. *Multiscale Modeling Simul.* **14**, 1153–1173 (2016).
- Rupp, M., Tkatchenko, A., Müller, K.-R. & von Lilienfeld, O. A. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.* **108**, 058301 (2012).
- Balabin, R. M. & Lomakina, E. I. Support vector machine regression (LS-SVM)—an alternative to artificial neural networks (ANNs) for the analysis of quantum chemistry data? *Phys. Chem. Chem. Phys.* **13**, 11710–11718 (2011).
- Seko, A., Takahashi, A. & Tanaka, I. First-principles interatomic potentials for ten elemental metals via compressed sensing. *Phys. Rev. B* **92**, 054113 (2015).
- Brown, W. M., Thompson, A. P. & Schultz, P. A. Efficient hybrid evolutionary optimization of interatomic potential models. *J. Chem. Phys.* **132**, 24108 (2010).
- Mueller, T. & Ceder, G. Bayesian approach to cluster expansions. *Phys. Rev. B* **80**, 024103 (2009).
- Behler, J. Perspective machine learning potentials for atomistic simulations. *J. Chem. Phys.* **145**, 170901 (2016).

12. Chmiela, S. et al. Machine learning of accurate energy-conserving molecular force fields. *Sci. Adv.* **3**, e1603015 (2017).
13. Li, Z., Kermode, J. R. & De Vita, A. Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces. *Phys. Rev. Lett.* **114**, 096405 (2015).
14. Botu, V. & Ramprasad, R. Adaptive machine learning framework to accelerate ab initio molecular dynamics. *Int. J. Quantum Chem.* **115**, 1074–1083 (2015).
15. Artrith, N., Urban, A. & Ceder, G. Efficient and accurate machine-learning interpolation of atomic energies in compositions with many species. *Phys. Rev. B* **96**, 014112 (2017).
16. Smith, J. S., Isayev, O. & Roitberg, A. E. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chem. Sci.* **8**, 3192–3203 (2017).
17. Cao, L., Li, C. & Mueller, T. The use of cluster expansions to predict the structures and properties of surfaces and nanostructured materials. *J. Chem. Inf. Modeling* **58**, 2401–2413 (2018).
18. Zhang, L., Han, J., Wang, H., Car, R. & Weinan, E. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Phys. Rev. Lett.* **120**, 143001 (2018).
19. Yunxing, Z. et al. A performance and cost assessment of machine learning interatomic potentials. *arXiv:1906.08888v3 [physics.comp-ph]* (2019).
20. Nyshadham, C. et al. Machine-learned multi-system surrogate models for materials prediction. *npj Computational Mater.* **5**, 51 (2019).
21. Mueller, T., Kusne, A. G. & Ramprasad, R. in *Reviews in Computational Chemistry*. Vol. 29 (eds Parrill, A. L. & Lipkowitz, K. B.) (John Wiley & Sons, Inc. 2016).
22. Daw, M. S. & Baskes, M. I. Embedded-atom method: derivation and application to impurities, surfaces, and other defects in metals. *Phys. Rev. B* **29**, 6443–6453 (1984).
23. Finnis, M. W. & Sinclair, J. E. A simple empirical N-body potential for transition metals. *Philos. Mag. A* **50**, 45–55 (1984).
24. Ercolessi, F., Parrinello, M. & Tosatti, E. Simulation of gold in the glue model. *Philos. Mag. A* **58**, 213–226 (1988).
25. Brenner, D. W., Shenderova, O. A. & Areshkin, D. A. Quantum-based analytic interatomic forces and materials simulation. *Rev. Computational Chem.* <https://doi.org/10.1002/9780470125892.ch4> (1998).
26. Tersoff, J. New empirical approach for the structure and energy of covalent systems. *Phys. Rev. B* **37**, 6991–7000 (1988).
27. Sinnott, S. B. & Brenner, D. W. Three decades of many-body potentials in materials research. *MRS Bull.* **37**, 469–473 (2012).
28. Finnis, M. W. Concepts for simulating and understanding materials at the atomic scale. *MRS Bull.* **37**, 477–484 (2012).
29. Plimpton, S. J. & Thompson, A. P. Computational aspects of many-body potentials. *MRS Bull.* **37**, 513–521 (2012).
30. Chan, H. et al. Machine learning classical interatomic potentials for molecular dynamics from first-principles training data. *J. Phys. Chem. C* **123**, 6941–6957 (2019).
31. Li, Y. et al. Machine learning force field parameters from ab initio data. *J. Chem. Theory Comput.* **13**, 4492–4503 (2017).
32. Pahari, P. & Chaturvedi, S. Determination of best-fit potential parameters for a reactive force field using a genetic algorithm. *J. Mol. Modeling* **18**, 1049–1061 (2012).
33. Larsson, H. R., van Duin, A. C. T. & Hartke, B. Global optimization of parameters in the reactive force field ReaxFF for SiOH. *J. Comput. Chem.* **34**, 2178–2189 (2013).
34. Sen, F. G. et al. Towards accurate prediction of catalytic activity in IrO₂ nanoclusters via first principles-based variable charge force field. *J. Mater. Chem. A* **3**, 18970–18982 (2015).
35. Cherukara, M. J. et al. Ab initio-based bond order potential to investigate low thermal conductivity of stanene nanostructures. *J. Phys. Chem. Lett.* **7**, 3752–3759 (2016).
36. Koza, J. R. *Genetic programming—on the programming of computers by means of natural selection*. (MIT Press, 1992).
37. Tim Mueller, A. G. K. & Ramprasad, R. in *Reviews in Computational Chemistry*. Vol. 29 (eds Parrill, Abby L. & Lipkowitz, Kenny B.) Ch. 4, 188 (John Wiley & Sons, Inc., 2016).
38. Schmidt, M. & Lipson, H. Distilling free-form natural laws from experimental data. *Science* **324**, 81–85 (2009).
39. Mueller, T., Jöhlin, E. & Grossman, J. C. Origins of hole traps in hydrogenated nanocrystalline and amorphous silicon revealed through machine learning. *Phys. Rev. B* **89**, 115202 (2014).
40. Yuan, F. & Mueller, T. Identifying models of dielectric breakdown strength from high-throughput data via genetic programming. *Sci. Rep.* **7**, 17594 (2017).
41. Slepoy, A., Peters, M. D. & Thompson, A. P. Searching for globally optimal functional forms for interatomic potentials using genetic programming with parallel tempering. *J. Comput. Chem.* **28**, 2465–2471 (2007).
42. Abdel Kenoufi, K. T. K. Symbolic regression of inter-atomic potentials via genetic programming. *Biol. Chem. Res.* **2**, 1–15 (2015).
43. Makarov, D. E. & Metiu, H. Fitting potential-energy surfaces: a search in the function space by directed genetic programming. *J. Chem. Phys.* **108**, 590–598 (1998).
44. Lennard-Jones, J. E. Cohesion. *Proc. Phys. Soc.* **43**, 461–482 (1931).
45. Coulomb, C. A. *Mémoires sur l'électricité et la magnétisme*. (Chez Bachelier, libraire, 1789).
46. Morse, P. M. Diatomic molecules according to the wave mechanics. *li. Vibrational Lev. Phys. Rev.* **34**, 57–64 (1929).
47. Brenner, D. W. Relationship between the embedded-atom method and Tersoff potentials. *Phys. Rev. Lett.* **63**, 1022–1022 (1989).
48. Abell, G. C. Empirical chemical pseudopotential theory of molecular and metallic bonding. *Phys. Rev. B* **31**, 6184–6196 (1985).
49. Hawkins, D. M. The problem of overfitting. *J. Chem. Inf. Computer Sci.* **44**, 1–12 (2004).
50. Ercolessi, F. & Adams, J. B. Interatomic potentials from first-principles calculations: the force-matching method. *Europhys. Lett. (EPL)* **26**, 583–588 (1994).
51. Cloutman, L. D. A selected library of transport coefficients for combustion and plasma physics applications, <https://doi.org/10.2172/793685> (2000).
52. Hohenberg, P. & Kohn, W. Inhomogeneous electron gas. *Phys. Rev.* **136**, B864–B871 (1964).
53. Chen, C. et al. Accurate force field for molybdenum by machine learning large materials data. *Phys. Rev. Mater.* **1**, 43603 (2017).
54. Mishin, Y., Mehl, M. J., Papaconstantopoulos, D. A., Voter, A. F. & Kress, J. D. Structural stability and lattice defects in copper: Ab initio, tight-binding, and embedded-atom calculations. *Phys. Rev. B* **63**, 224106 (2001).
55. Onat, B. & Durukanoğlu, S. An optimized interatomic potential for Cu–Ni alloys with the embedded-atom method. *J. Phys.: Condens. Matter* **26**, 035404 (2013).
56. Foiles, S. M., Baskes, M. I. & Daw, M. S. Embedded-atom-method functions for the fcc metals Cu, Ag, Au, Ni, Pd, Pt, and their alloys. *Phys. Rev. B* **33**, 7983–7991 (1986).
57. Wulff, G. Zur Frage der Geschwindigkeit des Wachstums und der Auflösung der Krystallflächen. *Z. Kryst. Mineral.* **34**, 449–530 (1901).
58. Mendeleev, M. I., Kramer, M. J., Becker, C. A. & Asta, M. Analysis of semi-empirical interatomic potentials appropriate for simulation of crystalline and liquid Al and Cu. *Philos. Mag.* **88**, 1723–1750 (2008).
59. Plimpton, S. Fast parallel algorithms for short-range molecular dynamics. *J. Computational Phys.* **117**, 1–19 (1995).
60. Wood, M. A., Thompson, A. P. & W., B. D. Extending the accuracy of the SNAP interatomic potential form. *J. Chem. Phys.* **148**, 241721 (2018).
61. Yao, K., Herr, J. E., Toth, D. W., McKintyre, R. & Parkhill, J. The TensorMol-0.1 model chemistry: a neural network augmented with long-range physics. *Chem. Sci.* **9**, 2261–2269 (2018).
62. Behler, J. in *Chemical Modelling: Applications and Theory*, Vol. 7. 1–41 (The Royal Society of Chemistry, 2010).
63. Deringer, V. L. & Csányi, G. Machine learning based interatomic potential for amorphous carbon. *Phys. Rev. B* **95**, 094203 (2017).
64. Behler, J. Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations. *Phys. Chem. Chem. Phys.* **13**, 17930–17955 (2011).
65. Szlachta, W. J., Bartók, A. P. & Csányi, G. Accuracy and transferability of Gaussian approximation potential models for tungsten. *Phys. Rev. B* **90**, 104108 (2014).
66. Artrith, N. & Behler, J. High-dimensional neural network potentials for metal surfaces: A prototype study for copper. *Phys. Rev. B* **85**, 045439 (2012).
67. Smits, G. F. & Kotanchek, M. in *Genetic Programming Theory and Practice II*. 283–299 (Springer, 2005).
68. Borges, C. E. et al. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*. 985–986 (ACM, Portland: Oregon, USA, 2010).
69. Ding, H.-Q., Karasawa, N. & Goddard, W. A. Optimal spline cutoffs for Coulomb and van der Waals interactions. *Chem. Phys. Lett.* **193**, 197–201 (1992).
70. Poli, R., Langdon, W. B. & McPhee, N. F. *A Field Guide to Genetic Programming*. (2008). Published via <http://lulu.com> and available at <http://www.gp-fieldguide.org.uk>.
71. Jeffrey Horn, N. N. & Goldberg, D. E. A niched Pareto genetic algorithm for multiobjective optimization. In *Proc. First IEEE Conf. Evolutionary Computation*. 82–87 (IEEE, 1994).
72. Ekárt, A. & Németh, S. Z. Selection based on the pareto nondomination criterion for controlling code growth in genetic programming. *Genet. Program. Evol. Mach.* **2**, 61–73 (2001).
73. Hansen, N. & Ostermeier, A. In *Proceedings of IEEE International Conference on Evolutionary Computation*. 312–317 (IEEE, 1996).
74. Polak, E. & Ribiere, G. Note sur la convergence des methodes de directions conjuguées. *Imform. Rech. Oper.* **16**, 35 (1969).
75. Kresse, G. & Furthmüller, J. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B* **54**, 11169 (1996).
76. Perdew, J. P., Burke, K. & Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **77**, 3865–3868 (1996).

77. Blöchl, P. E. Projector augmented-wave method. *Phys. Rev. B* **50**, 17953–17979 (1994).
78. Wisesa, P., McGill, K. A. & Mueller, T. Efficient generation of generalized Monkhorst-Pack grids through the use of informatics. *Phys. Rev. B* **93**, 155109 (2016).
79. Sutton, A. P. & Chen, J. Long-range Finnis–Sinclair potentials. *Philos. Mag. Lett.* **61**, 139–146 (1990).
80. Mendelev, M. I. & King, A. H. The interactions of self-interstitials with twin boundaries. *Philos. Mag.* **93**, 1268–1278 (2013).
81. Adams, J. B., Foiles, S. M. & Wolfer, W. G. Self-diffusion and impurity diffusion of fee metals using the five-frequency model and the Embedded Atom Method. *J. Mater. Res.* **4**, 102–112 (1989).
82. Becker, C. A., Tavazza, F., Trautt, Z. T. & Buarque de Macedo, R. A. Considerations for choosing and using force fields and interatomic potentials in materials science and engineering. *Curr. Opin. Solid State Mater. Sci.* **17**, 277–283 (2013).
83. Rose, J. H., Smith, J. R., Guinea, F. & Ferrante, J. Universal features of the equation of state of metals. *Phys. Rev. B* **29**, 2963–2969 (1984).

ACKNOWLEDGEMENTS

We acknowledge financial support from the Office of Naval Research, grant number N000141512665. This work was done using high-performance computing resources from the Maryland Advanced Research Computing Cluster (MARCC) and from the Homewood High-Performance Cluster (HHPC). We thank Qing-Jie Li, Zhao Fan, Dihui Ruan, and the members of our research group for useful discussions.

AUTHOR CONTRIBUTIONS

T.M. conceived of and managed the project. T.M. and A.H. developed the software and wrote the manuscript. A.H., A.B. and F.Y. computed the data. A.H. and A.B. worked on open sourcing the software. A.H. and S.A.M.M. ran experiments. A.B. implemented the models in LAMMPS. A.H. developed data analysis scripts. All the authors proposed, discussed, or developed ideas that improved the performance of the machine-learning algorithm or the quality of the data.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information is available for this paper at <https://doi.org/10.1038/s41524-019-0249-1>.

Correspondence and requests for materials should be addressed to T.M.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2019