A UNIVERSAL FRAMEWORK FOR FEATURIZATION OF ATOMISTIC SYSTEMS

A PREPRINT

Xiangyun Lei
School of Chemical and Biomolecular Engineering
Georgia Institute of Technology
Atlanta, GA, 30318 USA
xlei38@gatech.edu

Andrew J. Medford
School of Chemical and Biomolecular Engineering
Georgia Institute of Technology
Atlanta, GA, 30318 USA
ajm@gatech.edu

August 9, 2022

ABSTRACT

Molecular dynamics simulations are an invaluable tool in numerous scientific fields. However, the ubiquitous classical force fields cannot describe reactive systems, and quantum molecular dynamics are too computationally demanding to treat large systems or long timescales. Reactive force fields based on physics or machine learning can be used to bridge the gap in time and length scales, but these force fields require substantial effort to construct and are highly specific to a given chemical composition and application. A significant limitation of machine learning models is the use of element-specific features, leading to models that scale poorly with the number of elements. This work introduces the Gaussian multipole (GMP) featurization scheme that utilizes physically-relevant multipole expansions of the electron density around atoms to yield feature vectors that interpolate between element types and have a fixed dimension regardless of the number of elements present. We combine GMP with neural networks to directly compare it to the widely used Behler-Parinello symmetry functions for the MD17 dataset, revealing that it exhibits improved accuracy and computational efficiency. Further, we demonstrate that GMP-based models can achieve chemical accuracy for the QM9 dataset, and their accuracy remains reasonable even when extrapolating to new elements. Finally, we test GMP-based models for the Open Catalysis Project (OCP) dataset, revealing comparable performance to graph convolutional deep learning models. The results indicate that this featurization scheme fills a critical gap in the construction of efficient and transferable machine-learned force fields.

Atomistic simulations are a crucial tool in many scientific fields, ranging from protein engineering to materials design [1, 2, 3, 4, 5, 6, 7, 8, 9]. Full quantum mechanical treatment of atoms provides highly accurate energies and forces, but the computational cost is prohibitive for the length and time scales relevant to most applications [10, 11, 12]. Classical and reactive force fields can act as surrogates for the quantum mechanical simulations, enabling simulations at longer length and time scales [13, 14, 15, 16, 17, 18, 7, 19]. However, these models are specialized to specific systems and have a limited ability to simulate inherently quantum-mechanical phenomena such as covalent bond formation [20]. Machine-learning models have recently emerged as a promising strategy to fill the gap between quantum mechanical simulations and classical force field models [21, 22, 23]. The field of machine-learned force fields has exploded in the last decade, leading to a plethora of different machine-learning force field models capable of predicting energies and forces with accuracy comparable to the underlying method [24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38]. However, most models are customized for specific application domains, and a framework for a general-purpose model has not been established.

One fundamental problem in machine-learning models for atomistic systems is the issue of feature generation. Cartesian coordinates and elemental identities are the most common way to define atomistic systems. However, this description does not capture the system's rotational, translational, or permutation invariances. Converting the Cartesian coordinates to "feature vectors" that describe each atom is commonly used to solve this problem. Researchers have devised a wide range of featurization strategies that encode these fundamental physical symmetries, recently classified and thoroughly covered in several thorough reviews [39, 40]. Some examples include the Coulomb matrix[41], radial distribution

function-based fingerprint [42], FCHL[35], SOAP [36, 37], COMB [17], Chebyshev polynomials [43], Gaussian momentums [44] and the ubiquitous atom-centered symmetry functions[24]. These strategies have yielded remarkably accurate models within various sub-fields but are not sufficiently general to treat all atomistic systems.

The most common limitation is related to the scaling of the feature vector size as the number of elements in the system increases. Most existing descriptors scale polynomially or combinatorially with the number of elements in the system [24, 36]. This poor scaling means that these featurization strategies can only be applied to training sets with a limited number of elements, making the prospect of a universal machine-learning model that works for all elements infeasible. While some approaches overcome this limitation [38, 43], they have not been tested in on large many-element benchmark systems. In addition to fingerprint scalability, there is also the issue of scalability of the regression model. Many high-accuracy models rely on kernel ridge regression (KRR), a technique that suffers from poor scalability at with extremely large training set sizes. Neural networks are far more scalable, but if typical element-specific featurization schemes are used then a high-dimensional neural network (HDNN) [24, 45] is required to connect the features to atomic properties. The HDNN scheme uses element-specific neural networks, so that the number of model parameters also increases with the number of elements. An alternative to featurization is deep learning [46], typically with graph-convolutional neural network (GCN) models [25, 26, 27, 28, 32, 29, 30]. These deep learning models show an excellent ability to learn appropriate representations for molecular, solid-state, and surface systems with many elements [47]. However, this comes at the cost of less transparent models that typically require more time for training and prediction than their feature-based counterparts. Moreover, many of the deep learning approaches utilize elemental features [32, 25, 27, 28], suggesting that improved featurization schemes will translate to improved deep learning models.

Here we introduce a new featurization approach called Gaussian multipole (GMP) features. The GMP features utilize an implicit description of the electron density so that the feature vector's dimension is independent of the number of elements. Using the electron density as the fundamental input makes them suitable for universal machine-learning models that work for all elements and provides a straightforward route for extending them to systems that involve charged atoms or magnetic moments. It also naturally allows the use of the more efficient single neural network (SNN) [33] structure, where all elements share the same neural network. Moreover, the GMP features are related to a multipole expansion of the electron density [48, 49], making them physically relevant and systematically improvable. Remarkably, we show that the GMP features are capable of interpolating between elemental species, a capability that to our knowledge has not been demonstrated with any other featurization schemes. These properties of the GMP features will facilitate a new family of fast and interpretable feature-based machine learning models that have the general applicability of more complex and opaque GCN models.

The GMP approach encodes the elemental identity through an approximate description of the electron density based on Gaussian basis functions. In this work, we use the valence density extracted from the SG15 pseudopotentials [50] approximated by 2-7 atom-centered Gaussians per element, normalized by the total number of valence electrons. The number and widths of the Gaussians for each element are determined using nonlinear regression, and these "static valence densities" are fixed for all models presented in this work. Details are provided in the supporting information. Conceptually, this is equivalent to using the valence density of non-interacting atoms as the fundamental description of the system (Fig. 1a). This approach also allows for interpolation and extrapolation between different elemental species, and concepts like charged atoms can be naturally accommodated using the isolated charged atom to construct a suitable valence potential. In principle, it is also possible to use other quantities like the all-electron density, spin density, or self-consistent electron density, as long as the quantity can be represented by a linear combination of atom-centered Gaussians.

The approximated electron density is then vectorized via the inner product between the electron density and atom-centered "probe" functions to generate the features for the atoms. The probe function consists of a Gaussian function and a Maxwell-Cartesian spherical harmonic (MCSH) function. Gaussian functions of varying width account for the radial variation (Fig. 1b) of surrounding electron density of an atom, and the MCSH functions capture angular variation (Fig. 1c). The Gaussian product rule provides analytical solutions to the integral in Eqn. 1, enabling highly efficient computation of the features. Rotational invariance is enforced by taking norms of each MCSH order[51] (different than the previous publication [52]).

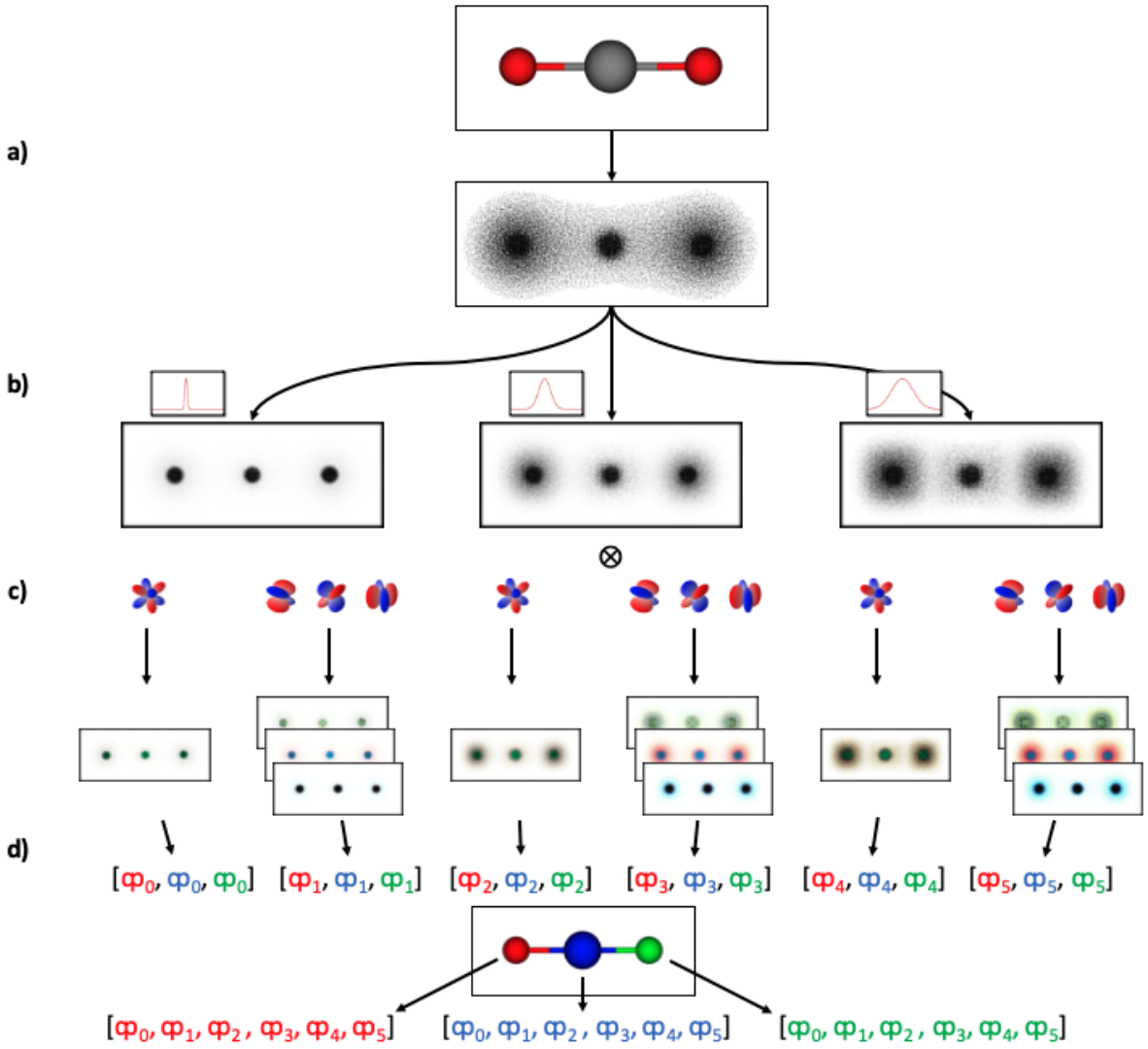


Figure 1: Illustration for the GMP featurization scheme. a) Construct the electron density distribution using linear combination of Gaussians. b) Apply the Gaussian radial probe to focus on the electronic environments at different radial length scales around each atom core by masking the electron density distribution with Gaussian functions of different widths. c) Apply the angular (MCSH) probe that acts as a multi-pole expansion of the radially-masked electron density via inner products with different groups of MCSH functions. d) Take the norm of the results for each group to ensure rotational invariance, yielding an entry to the feature vector for each individual atom.

Mathematically this is described as

$$
\begin{aligned}
\mu_{i,abc} &=< probe, \hat{\rho} >=< angular\ probe \times radial\ probe, \hat{\rho} > \\
&=< S_{abc} \times G_i, \sum_{\substack{j \\ atoms}} \sum_{\substack{k \\ Gaussians}} G_{dens,j,k} > \\
&= \iiint_V S_{abc} G_{probe,i} \sum_{\substack{j \\ atoms}} \sum_{\substack{k \\ Gaussians}} G_{dens,j,k} dV \\
&= \sum_{\substack{j \\ atoms}} \sum_{\substack{k \\ Gaussians}} \iiint_V S_{abc} G_{probe,i} G_{dens,j,k} dV
\end{aligned}
\tag{1}
$$

where $< a, b >$ denotes inner product of two functions, V is the volume, $\mu_{i,abc}$ is the feature resulting from the radial probe G_i and angular probe S_{abc}. $\hat{\rho}$ is the distribution of electron density of a molecule, approximated by linear combinations of primitive Gaussians $G_{dens,j,k}$ centered at each atom. The formalism does not include a strict cutoff radius, though for computational efficiency the sum can be restricted to surrounding atoms with a non-negligible overlap. The MCSH functions S_{abc} are linear combinations of polynomials:

$$S^n_{abc} = \sum_{\substack{t \\ terms}} C_t x^{m_{x,t}} y^{m_{y,t}} z^{m_{z,t}} \tag{2}$$

where abc is the specific index of the spherical harmonic function, $n = a + b + c$ is the order of it, and m_x, m_y, m_z are the exponents of the specific polynomial. The first 4 orders of MCSH functions are listed in Table 1.

n	group	{abc}	$S^{(n)}_{abc}$	n	group	{abc}	$S^{(n)}_{abc}$
0	1	000	1	3	1	300	$15x^3 - 9x$
1	1	100	x			030	$15y^3 - 9y$
		010	y			030	$15y^3 - 9y$
		001	z		2	210	$15x^2y - 3y$
2	1	200	$3x^2 - 1$			201	$15x^2z - 3z$
		020	$3y^2 - 1$			021	$15y^2z - 3z$
		002	$3z^2 - 1$			120	$15xy^2 - 3x$
	2	110	$3xy$			102	$15xz^2 - 3x$
		101	$3xz$			012	$15yz^2 - 3y$
		011	$3yz$		3	111	$15xyz$

Table 1: The analytical expressions of the first four orders of MCSH denoted by $S^{(n)}_{abc}$

To ensure that the resulting features are rotationally invariant, we use the weighted sum of square of each order, μ_i. Please note that this definition is different than our previous publication [52]. Therefore, the GMP feature vector is defined as

$$\vec{\varphi} = \varphi_i = \sqrt{w_{abc} \sum\nolimits_{P(a,b,c)} \mu^2_{i,abc}} \mid a, b, c \in \mathbb{N}, \tag{3}$$

where φ denotes the GMP features, i is an index over the radial probes, abc is an index combination corresponding to a rotational group, and $P(a, b, c)$ denotes the permutation group of a, b, c (e.g. $P(1, 0, 0) = \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$). The weight w_{abc} is shared among MCSH within the specific subgroup (perturbaion group) of order $n = a + b + c$

$$w_{abc} = \frac{n!}{a!b!c!} \tag{4}$$

Therefore, the set of features can thus be written as

$$\vec{\varphi} = \left\{ \begin{array}{l} \sqrt{\mu^2_{1,000}}, \sqrt{\mu^2_{1,100} + \mu^2_{1,010} + \mu^2_{1,001}}, \sqrt{(\mu^2_{1,200} + \mu^2_{1,020} + \mu^2_{1,002}) + 2.0(\mu^2_{1,110} + \mu^2_{1,101} + \mu^2_{1,011})}, \dots \\ \sqrt{\mu^2_{2,000}}, \sqrt{\mu^2_{2,100} + \mu^2_{2,010} + \mu^2_{2,001}}, \sqrt{(\mu^2_{2,200} + \mu^2_{2,020} + \mu^2_{2,002}) + 2.0(\mu^2_{2,110} + \mu^2_{2,101} + \mu^2_{2,011})}, \dots \\ \dots \end{array} \right\}. \tag{5}$$

Conceptually, the resulting features are similar to a multipole expansion of the electron density around each atom. The multipole expansion provides rotational features that are complete and orthogonal. The Gaussian probe's width controls the radial length scale of the multi-pole expansions, and the radial features are over-complete. These properties reduce linear dependencies within the features and lead to a systematic improvement in the system's description as the number of features increases. The GMP scheme is closely related to the "wavelet" features [38], but it uses a different representation of the electron density, and differs in the way the spherical harmonics are normalized and computed, so that it provides atom-centered features. It is also similar to the SOAP scheme, which uses element-specific densities and spherical harmonics [24, 36, 37]. However, the key difference is that the length of the feature vector is fixed regardless

of the number elements, and the normalization across rotational groups leads to fewer features. These enable GMP to efficiently featurize complex systems with an arbitrary number of elements and arbitrary boundary conditions.

In this work, we combine the GMP features with a neural network regression model to demonstrate the efficiency, accuracy, and transferability of models based on these features. As mentioned, the single neural network (SNN) architecture [33] is well-suited to the GMP features since all elements utilize the same features (Fig. 1). We also utilize a per-element bias term, equivalent to fitting to formation energies instead of total energies, to reduce the magnitude of the energies. We have implemented the GMP+SNN framework in the AMPTorch code [53, 54] and used this implementation for all models in this work. The number of parameters used by the GMP+SNN model varies depending on the number of features, hidden layers, and nodes per layer but is generally lower than the number of parameters in a comparable GCN by an an order of magnitude. Details of model fitting, implementation, and parameters are provided in the supplementary information. The notation GMP($N_{Gaussian}$,N_{MCSH}) is used to denote the feature sets in the examples below, where $N_{Gaussian}$ is the number of radial Gaussians, and N_{MCSH} is the maximum order MCSH used to construct the feature set. The notation SNN($[list\ layers]$) is used to denote the SNN model, The activation function is $Tanh$ for the MD17 test, and $GELU$ for the QM9 and OC20 tests, and batch normalization is always applied for each layer. Therefore, GMP(9,3) is a feature set constructed using 9 radial Gaussians, resulting in $9 \times 4 = 36$ features and GMP(9,3)+SNN(50,50,50) is a SNN model based on the GMP(9,3) feature set with 3 hidden layers and 50 nodes per layer. We note that models with the same number of radial Gaussian probes are not necessarily equivalent, since Gaussian widths are determined manually at this point. Additional details including the widths used for the radial Gaussians are provided in the supplementary information.

First, we compare the GMP+SNN model to the Behler-Parinello neural network (BPNN) approach that is one of the most computationally efficient featurization schemes [39] and ubiquitous in materials science and chemistry due to its simplicity, generality, and efficiency [24, 55, 56, 57, 58, 59, 54]. We utilize an established molecular dynamics trajectory of the 3-element (C, H, O) aspirin molecule at the DFT/PBE+vdW-TS level of theory for this comparison[60]. For BPNN featurization, we use 12 variants inspired by examples in literature [61]. All neural networks consist of 3 hidden layers with 50 nodes each, and the BPNN approach uses separate neural networks for each element (resulting in 3 times the number of parameters). We use a training set size of 40K images for all models, and the test and validation sets each contain 10K images. We ensure robustness by using ten randomly selected train/test/validation sets. The performance is measured by the mean absolute error (MAE) for the predicted energies of the test set images.

Fig 2a shows the GMP+SNN model error as a function of the multipole expansion order and the number of Gaussian radial probes. The results reveal that the GMP+SNN accuracy increases systematically with the multipole expansion order and the number of radial probes, providing a clear strategy for identifying an appropriate feature set for a given problem. In Fig 2b, we compare the accuracy of the GMP+SNN model to the BPNN model as a function of the number of features. All Pareto-optimal models utilize the GMP+SNN approach, despite the fact that the BPNN models have three times more fitted parameters due to element-specific neural networks. It is also clear that the accuracy of BPNN models does not systematically improve with the number of features. Finally, we compare the wall time needed to compute a single image for each model. To ensure a fair comparison, we use the same CPU for each test, both approaches use the same C++ source code and loop structure, and the time is on average over the 10K predicted validation images (see supplementary information). Fig. 2c) shows that the GMP+SNN framework is always faster than the BPNN approach at a fixed accuracy level, or is always more accurate for a fixed computation time. This example demonstrates that the GMP+SNN approach is capable of achieving a lower error than the BPNN approach with fewer parameters and less time.

The performance of the GMP+SNN can be further improved with force regularization, with a loss function defined as

$$\Gamma = \frac{1}{N_{image}} \sum_{i=1}^{N_{image}} \left\{ \left\| E_{NN}^{i} - E_{ref}^{i} \right\| + \frac{\beta}{3N_{atom}^{i}} \sum_{j=1}^{3N_{atom}^{i}} \left\| F_{j,NN}^{i} - F_{j,ref}^{i} \right\| \right\}, \tag{6}$$

where N_{image} is the number of training images, N_{atom}^{i} is the number of atoms in image i, E_{NN} and E_{ref} are the predicted and reference energy of a image, F_{NN} and F_{ref} are the predicted and reference forces of an atom along each Cartesian axis, and β is the force regularization coefficient. Figure 3 shows the results of GMP+SNN with 2 feature sets, GMP(6,3)+SNN(50,50,50) and GMP(10,6)+SNN(50,50,50), as a function of the force regularization coefficient, β. These two models are both on the Pareto frontier from the previous test. For simplicity, this test was done with 1K training images and 1K test images. The results confirm that energy training benefits from force regularization. In this case, the energy error is reduced by a factor of 2 to 3 at the optimal regularization coefficient. Both energy and force prediction accuracy show the same trend and the same optimal force coefficient, indicating that the model avoids the apparent trade-off between energy and force accuracy that has been observed for some GCN model architectures [47].

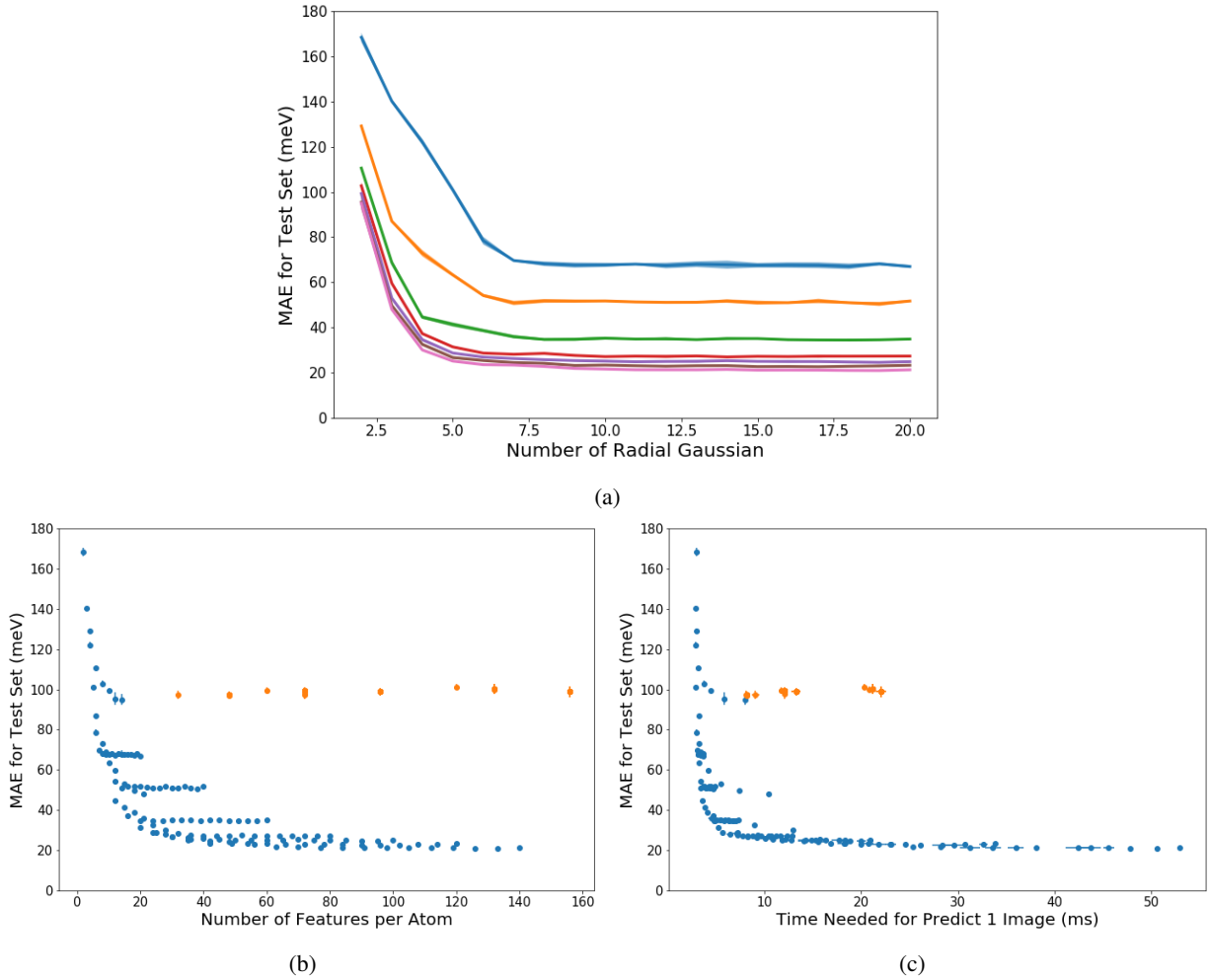


Figure 2: Training results of GMP+SNN models compared to Behler-Parrinello + HDNN models. The training set consists of 40K images, validation set consists of 10K images and test set consists of 10K images, all randomly drawn from the MD17 aspirin simulation trajectory. Each setting was tested 10 times to ensure robustness. a) Convergence of test set MAE as rotational order and number of radial Gaussians are increased, with curves correspond to angular MCSH probes up to 1 (orange), 2 (green), 3 (red), 4 (purple), 5 (brown) and 6 (pink) orders. b) Comparison of test MAE for GMP+SNN (blue) and BPNN (orange) models as a function of the number of features. c) Comparison of test MAE for GMP+SNN and BPNN models as a function of the average prediction time for a single image.

Next, we show that the GMP+SNN technique can scale to systems with more elements by training it on the atomization energy of the established QM9 benchmark dataset [62]. This dataset consists of 130K chemical species with up to 9 heavy atoms and five elements (C, H, O, N, F) optimized at the B3LYP level of theory. In this example, 4 GMP descriptor sets and corresponding SNNs of different sizes are trained and tested using energies of each system. The learning curves showing the out-of-sample test error [63] as a function of training set size are shown in Fig. 4a. The errors of the tested GMP+SNN models are just slightly higher than the best state-of-the-art models based on Gaussian processes or GCNs (about 6 meV with 100K training data) [63, 30, 31, 27, 25, 28, 35, 36], but the GMP+SNN models utilize fewer adjustable parameters and are more scalable than non-parametric models like Gaussian process regression.

Fig. 4b shows the results of transfer learning between different elemental species in the QM9 data set. We trained a model on the set of all molecules that do not contain fluorine (128,908 molecules), and tested the same model on fluorine-containing molecules (1,923 molecules). The model error increases by around one order of magnitude when the predictions include elements not present in the training set. However, the test set MAE is 0.32 eV, which is competitive with the accuracy of generalized gradient approximation of density functional theory (GGA DFT) [64]. This reveals that the model is reasonably accurate even for elements outside the training set. The error distribution for transfer learning is bimodal, which is primarily related to the number of F atoms in the test systems. If the error is normalized by the

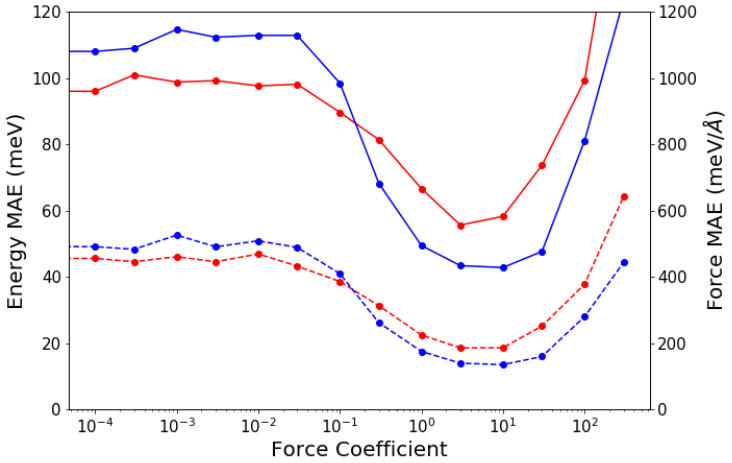
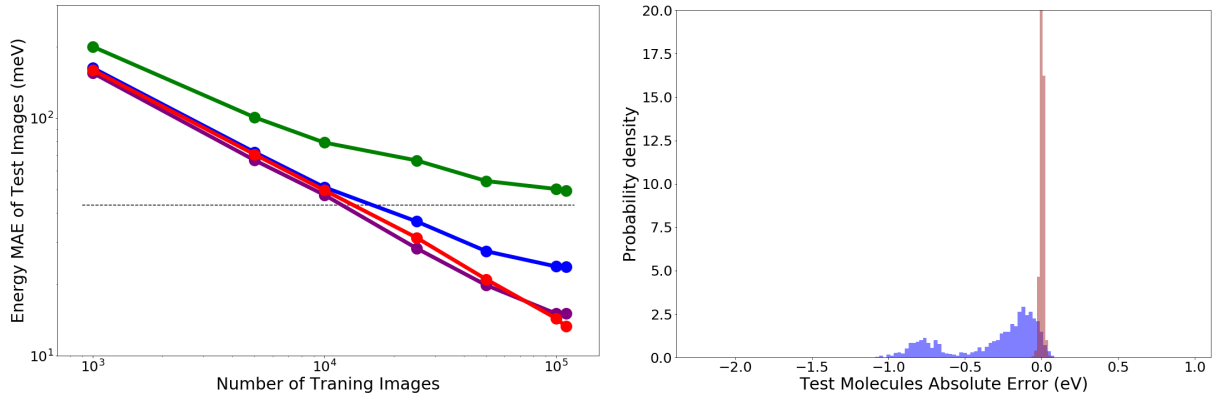


Figure 3: Test MAE of energies (solid) and forces (dashed) as a function of the force regularization coefficient on 1K randomly selected images from the aspirin trajectory. Two feature sets are compared here, both with SNN(50,3) architecture: GMP(6,3) (red, 24 features) and GMP(10,6) blue, 70 features).



(a) (b)

Figure 4: Learning curve and elemental transferability of GMP+SNN on the QM9 dataset. a) Learning curves of the GMP+SNN models with various feature sets and neural net sizes (green = GMP(10,1)+SNN(32,32,32), 20 features, blue = GMP(30,2)+SNN(64,64,64), 90 features, purple = GMP(50,4)+SNN(256,128,64), 250 features, red = GMP(70,6)+SNN(512,256,128,64), 490 features). The dashed line indicates chemical accuracy (43 meV). b) Test MAE distribution for GMP(70,6)+SNN(512,256,128,64) model trained exclusively on molecules without F and tested on all F-containing molecules (blue shade) and test MAE distribution for model trained and tested with randomly selected molecules (red shade).

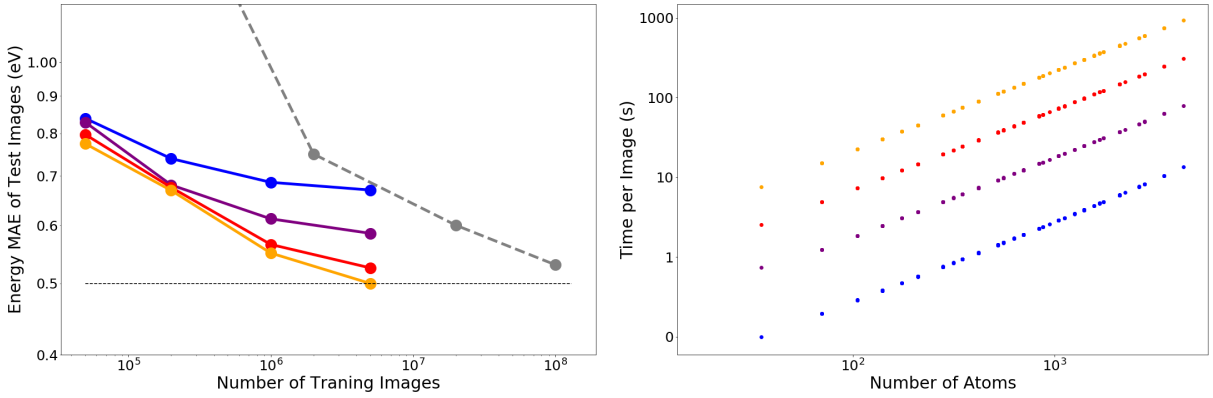
number of F atoms in the system the MAE decreases to 0.179 eV/F atom, and the distribution becomes more symmetric (see SI). In addition, we note that the error distribution for F atoms is relatively sensitive to the initial weights of the neural network. The MAE varies from 0.2 - 0.4 eV per system depending on initial weights(see SI). This indicates that the model is extrapolating and therefore more sensitive to the initial conditions and training procedure. However, the resulting models never fail catastrophically since the errors are always competitive with GGA DFT, indicating the general robustness of the transfer learning between elements.

Finally, we test the performance of the GMP+SNN approach on the Open Catalysis Project (OC20) S2EF dataset. This dataset consists of $>$100M geometries and corresponding energies of $>$100K adsorbate-catalyst pairs containing a total of 56 elements across 82 adsorbates and up to $> 100K$ different catalyst compositions for each adsorbate. The dataset also provides an independent set of 1M test systems [47]. The energies correspond to the GGA DFT level of theory with the RPBE functional [65], with mixed boundary conditions. The size, number of elements, and mix

of solid-state and molecular systems make this one of the most challenging benchmark datasets available. To date, the only models capable of training and prediction for this dataset utilize elaborate GCN models [47]. Fig. 5a shows the learning curves for four GMP+SNN models of different sizes tested on the provided in-domain (ID) validation set. The small model uses the GMP(30,2)+SNN(128,64,64) architecture (91 features, 34K parameters), the medium model uses the GMP(50,4)+SNN(256,128,64) architecture (250 features, 146K parameters), and the large model uses the GMP(70,6)+SNN(512,256,128,64) architecture (490 features, 528K parameters), and the largest model uses the GMP(90,8)+SNN(1024,512,128,64) architecture (810 features, 1.43M parameters). Details of all models are provided in the supplementary information. The full OC20 training set of 100M energies would require months to train with current computing resources available to us, so we restrict the analysis to energy training on data sets with fewer than 5M training images.

The results of training and testing on the OC20 set are shown in Fig. 5a. The in-domain test error reaches a minimum of 0.50 eV with 5M training images with the GMP(90,8)+SNN(1024,512,128,64) model, an error lower than all GCN models with 5M training points, and comparable to the performance of the CGCNN (0.527 eV) and DimeNet++ (0.486 eV) GCN models trained on all >100M training points [47]. Moreover, the GMP+SNN models require fewer parameters than the GCN models, with the largest GMP+SNN model having ∼1.4M parameters, compared to 1.8M (DimeNet++), 3.6M (CGCNN), and 7.4M (Schnet) parameters for the GCN models[47]. The error also decreases as the number of data points, the number of GMP features, or the neural net size increase. This suggests that further improvement is possible, although significant computational resources will be required to optimize and evaluate GMP+SNN models for the OC20 dataset.

Finally, the scaling of computational time versus the number of atoms in the system is plotted in Fig. 5b. The systems of different sizes are generated by making supercells that are periodic replications of an original unit cell of 35 atoms. The scaling is strictly linear due to the local nature of the features, which presents an advantage over most graph-based models require the entire structure as input. The linear scaling of the GMP+SNN model, along with a structure that is straightforward to parallelize, makes it ideal for the simulation of large systems that may not be feasible with graph-based models.



(a) (b)

Figure 5: Results for the OC20 dataset with GMP(30,2)+SNN(128,64,64) (blue, 90 features), GMP(50,4)+SNN(256,128,64) (purple, 250 features), GMP(70,6)+SNN(512,256,128,64) (red, 490 features), and GMP(90,8)+SNN(1024,512,128,64) (red, 810 features) models. a) Learning curves tested on the provided in-domain (ID) validation set with the DimeNet++ (gray dashed) results for comparison [47] (the gray dotted line at 0.5 eV represents an accuracy that would generally be considered useful) b) Log-log plot of time required to infer one images (with one core on a typical desktop) v.s. the number of atoms present. Systems of different sizes are made by periodically repeating one unit cell of C_2O adsorbed on Sn_3Ca (4 elements, 35 atoms).

These examples demonstrate that the GMP featurization scheme is an efficient and universal approach to fingerprinting atomistic systems with an arbitrary number of elements or atom types. The GMP features are more computationally efficient than the deep-learning GCN models that are commonly used for many-element systems, and are even faster than the widely used Behler-Parinello symmetry functions, resulting in machine-learned force fields that can be scaled to large many-element systems and long time scales. The GMP feature vectors utilize an implicit description of the system's electron density, making the number of features independent of the number of elements, facilitating the inclusion of general concepts from electronic structure theory, and enabling extrapolation to new element types. Moreover, the GMP features utilize physically-meaningful concepts, leading to more interpretable models that can be

systematically improved and providing a foundation for hybrid models that incorporate more physics. For example, the self-consistent electron density from a simplified Hamiltonian could be used as input to the GMP model, or the limiting behavior derived from electrostatics could be used to constrain the regression model.

The examples presented here use the SNN regression model in conjunction with the GMP features based on static valence density to obtain accuracy comparable to state-of-the-art models on the MD17, QM9, and OC20 datasets. The results confirm that the GMP+SNN approach can reach competitive accuracy with GCN based models despite having far fewer adjustable parameters and a simpler structure. There are numerous opportunities to revise and optimize the details of the GMP+SNN approach presented here, including modifying the valence density representation, optimizing feature selection and systematic optimization of the SNN architecture. However, the encouraging initial results across a wide range of application domains suggest that the GMP+SNN approach is a promising universal route to constructing efficient and general machine-learning models for atomistic systems.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences Computational Chemical Sciences program under Award Numbers DE-SC0019441 and DE-SC0019410. Computational effort was supplied partially by the National Science Foundation under Grant No. MRI-1828187. We acknowledge helpful discussions with Andrew Peterson, Zachary Ulissi, and Muhammed Shuaibi.

References

[1] Kieron Burke. Perspective on density functional theory. *The Journal of Chemical Physics*, 136(15):150901, 2012.

[2] Axel D. Becke. Perspective: Fifty years of density-functional theory in chemical physics. *The Journal of Chemical Physics*, 140(18):18A301, 2014.

[3] Narbe Mardirossian and Martin Head-Gordon. Thirty years of density functional theory in computational chemistry: an overview and extensive assessment of 200 density functionals. *Molecular physics*, 115(19):2315–2372, 2017.

[4] Adam Hospital, Josep Ramon Goñi, Modesto Orozco, and Josep L Gelpí. Molecular dynamics simulations: advances and applications. *Advances and applications in bioinformatics and chemistry*, 8:37–47, 2015.

[5] Marco De Vivo, Matteo Masetti, Giovanni Bottegoni, and Andrea Cavalli. Role of molecular dynamics and related methods in drug discovery. *Journal of medicinal chemistry*, 59(9):4035–4061, 2016.

[6] J. Andrew McCammon and Martin Karplus. Molecular dynamics simulations of biomolecules. *Nature structural biology*, 9(9):646–652, 2002.

[7] Judith A Harrison, J. David Schall, Sabina Maskey, Paul T Mikulski, M. Todd Knippenberg, and Brian H Morrow. Review of force fields and intermolecular potentials used in atomistic computational materials research. *Applied Physics Reviews*, 5(3):31104, 2018.

[8] Artem R Oganov, Chris J Pickard, Qiang Zhu, and Richard J Needs. Structure prediction drives materials discovery. *Nature reviews. Materials*, 4(5):331–348, 2019.

[9] Xinguo Ren, Patrick Rinke, Christian Joas, and Matthias Scheffler. Random-phase approximation and its applications in computational chemistry and materials science. *Journal of materials science*, 47(21):7447–7471, 2012.

[10] Edoardo Mosconi, Jon M Azpiroz, and Filippo De Angelis. Ab initio molecular dynamics simulations of methylammonium lead iodide perovskite degradation by water. *Chemistry of materials*, 27(13):4885–4892, 2015.

[11] Dominik Marx and Jürg Hutter. *Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods*. Cambridge University Press, 2009.

[12] Radu Iftimie, Peter Minary, Mark E. Tuckerman, and Bruce J. Berne. Ab initio molecular dynamics: Concepts, recent developments, and future trends. *Proceedings of the National Academy of Sciences - PNAS*, 102(19):6654–6659, 2005.

[13] B. R Brooks, C. L Brooks, A. D Mackerell, L Nilsson, R. J Petrella, B Roux, Y Won, G Archontis, C Bartels, S Boresch, A Caflisch, L Caves, Q Cui, A. R Dinner, M Feig, S Fischer, J Gao, M Hodoscek, W Im, K Kuczera, T Lazaridis, J Ma, V Ovchinnikov, E Paci, R. W Pastor, C. B Post, J. Z Pu, M Schaefer, B Tidor, R. M Venable, H. L Woodcock, X Wu, W Yang, D. M York, and M Karplus. Charmm: The biomolecular simulation program. *Journal of computational chemistry*, 30(10):1545–1614, 2009.

[14] David A Case, Thomas E Cheatham, Tom Darden, Holger Gohlke, Ray Luo, Kenneth M Merz, Alexey Onufriev, Carlos Simmerling, Bing Wang, and Robert J Woods. The amber biomolecular simulation programs. *Journal of computational chemistry*, 26(16):1668–1688, 2005.

[15] Adri C. T van Duin, Siddharth Dasgupta, Francois Lorant, and William A Goddard. Reaxff: A reactive force field for hydrocarbons. *The journal of physical chemistry. A, Molecules, spectroscopy, kinetics, environment, and general theory*, 105(41):9396–9409, 2001.

[16] William L Jorgensen and Julian Tirado-Rives. The opls [optimized potentials for liquid simulations] potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin. *Journal of the American Chemical Society*, 110(6):1657–1666, 1988.

[17] Simon R. Phillpot, Andrew C. Antony, Linyuan Shi, Michele L. Fullarton, Tao Liang, Susan B. Sinnott, Yongfeng Zhang, and S. Bulent Biner. Charge optimized many body (comb) potentials for simulation of nuclear fuel and clad. *Computational Materials Science*, 148:231 – 241, 2018.

[18] Thomas P Senftle, Sungwook Hong, Md Mahbubul Islam, Sudhir B Kylasa, Yuanxia Zheng, Yun Kyung Shin, Chad Junkermeier, Roman Engel-Herbert, Michael J Janik, Hasan Metin Aktulga, Toon Verstraelen, Ananth Grama, and Adri C T van Duin. The reaxff reactive force-field: development, applications and future directions. *npj computational materials*, 2(1):15011, 2016.

[19] Alexander D MacKerell. Chapter 7 empirical force fields for proteins: Current status and future directions. *Annual Reports in Computational Chemistry*, 1:91–102, 2005.

[20] Judith A Harrison, J. David Schall, Sabina Maskey, Paul T Mikulski, M. Todd Knippenberg, and Brian H Morrow. Review of force fields and intermolecular potentials used in atomistic computational materials research. *Applied Physics Reviews*, 5(3):31104, 2018.

[21] Chris M Handley and Paul L. A Popelier. Potential energy surfaces fitted by artificial neural networks. *The journal of physical chemistry. A, Molecules, spectroscopy, kinetics, environment, and general theory*, 114(10):3371–3383, 2010.

[22] Jörg Behler. Perspective: Machine learning potentials for atomistic simulations. *The Journal of chemical physics*, 145(17):170901–170901, 2016.

[23] Rampi Ramprasad, Rohit Batra, Ghanshyam Pilania, Arun Mannodi-Kanakkithodi, and Chiho Kim. Machine learning in materials informatics: recent applications and prospects. *npj computational materials*, 3(1):1–13, 2017.

[24] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.*, 98:146401, 4 2007.

[25] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 991–1001. Curran Associates, Inc., 2017.

[26] Tian Xie and Jeffrey C Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical review letters*, 120(14):145301–145301, 2018.

[27] Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2020.

[28] Oliver T Unke and Markus Meuwly. Physnet: A neural network for predicting energies, forces, dipole moments, and partial charges. *Journal of chemical theory and computation*, 15(6):3678–3693, 2019.

[29] Nicholas Lubbers, Justin S Smith, and Kipton Barros. Hierarchical modeling of molecular energies using a deep neural network. *The Journal of chemical physics*, 148(24):241715–241715, 2018.

[30] Shuo Zhang, Yang Liu, and Lei Xie. Molecular mechanics-driven graph neural network with multiplex graph for molecular structures, 2020.

[31] Z. Shui and G. Karypis. Heterogeneous molecular graph neural networks for predicting molecule properties. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 492–500, Los Alamitos, CA, USA, 2020. IEEE Computer Society.

[32] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. 2017.

[33] Mingjie Liu and John R. Kitchin. Singlenn: Modified behler–parrinello neural network with shared weights for atomistic simulations with transferability. *The Journal of Physical Chemistry C*, 124(32):17811–17818, 2020.

[34] J. S. Smith, O. Isayev, and A. E. Roitberg. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chem. Sci.*, 8:3192–3203, 2017.

[35] Felix A Faber, Anders S Christensen, Bing Huang, and O. Anatole von Lilienfeld. Alchemical and structural distribution based representation for universal quantum machine learning. *The Journal of chemical physics*, 148(24):241717–241717, 2018.

[36] Albert P Bartók, Sandip De, Carl Poelking, Noam Bernstein, James R Kermode, Gábor Csányi, and Michele Ceriotti. Machine learning unifies the modeling of materials and molecules. *Science advances*, 3(12):e1701816–e1701816, 2017.

[37] Miguel A. Caro. Optimizing many-body atomic descriptors for enhanced computational performance of machine learning based interatomic potentials. *Phys. Rev. B*, 100:024112, Jul 2019.

[38] Michael Eickenberg, Georgios Exarchakis, Matthew Hirn, and Stephane Mallat. Solid harmonic wavelet scattering: Predicting quantum molecular energy from invariant descriptors of 3d electronic densities. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[39] Marcel F. Langer, Alex Goeßmann, and Matthias Rupp. Representations of molecules and materials for interpolation of quantum-mechanical simulations via machine learning. *arXiv*, 2020.

[40] Felix Musil, Andrea Grisafi, Albert P. Bartók, Christoph Ortner, Gábor Csányi, and Michele Ceriotti. Physics-inspired structural representations for molecules and materials. *Chemical Reviews*, 121(16):9759–9815, jul 2021.

[41] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical review letters*, 108(5):058301–058301, 2012.

[42] Venkatesh Botu and Rampi Ramprasad. Adaptive machine learning framework to accelerate ab initio molecular dynamics. *International journal of quantum chemistry*, 115(16):1074–1083, 2015.

[43] Nongnuch Artrith, Alexander Urban, and Gerbrand Ceder. Efficient and accurate machine-learning interpolation of atomic energies in compositions with many species. *Phys. Rev. B*, 96:014112, Jul 2017.

[44] V. Zaverkin and J. Kästner. Gaussian moments as physically inspired molecular descriptors for accurate and scalable machine learning potentials. *Journal of Chemical Theory and Computation*, 16(8):5410–5421, 2020. PMID: 32672968.

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

[46] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and Weinan E. Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics. *Physical Review Letters*, 120(14), apr 2018.

[47] Lowik Chanussot, Abhishek Das, Siddharth Goyal, Thibaut Lavril, Muhammed Shuaibi, Morgane Riviere, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, Aini Palizhati, Anuroop Sriram, Brandon Wood, Junwoong Yoon, Devi Parikh, C. Lawrence Zitnick, and Zachary Ulissi. Open catalyst 2020 (OC20) dataset and community challenges. *ACS Catalysis*, 11(10):6059–6072, may 2021.

[48] A. R Edmonds. *Angular momentum in quantum mechanics.* Investigations in physics ; 4. 1957.

[49] William J Thompson and LeRoy F Cook. Angular momentum: An illustrated guide to rotational symmetries for physical systems. *American Journal of Physics*, 63(7):670–671, 1995.

[50] D. R. Hamann. Optimized norm-conserving vanderbilt pseudopotentials. *Phys. Rev. B*, 88:085117, Aug 2013.

[51] Jon Applequist. Maxwell–cartesian spherical harmonics in multipole potentials and atomic orbitals. *Theoretical Chemistry Accounts*, 107(2):103–115, 2002.

[52] Xiangyun Lei and Andrew J. Medford. Design and analysis of machine learning exchange-correlation functionals via rotationally invariant convolutional descriptors. *Phys. Rev. Materials*, 3:063801, 6 2019.

[53] Amptorch. https://github.com/ulissigroup/amptorch, 2020.

[54] Muhammed Shuaibi, Saurabh Sivakumar, Rui Qi Chen, and Zachary W Ulissi. Enabling robust offline active learning for machine learning potentials using simple physics-based priors. *Machine Learning: Science and Technology*, 2(2), 2020.

[55] Kyuhyun Lee, Dongsun Yoo, Wonseok Jeong, and Seungwu Han. Simple-nn: An efficient package for training and executing neural-network interatomic potentials. *Computer Physics Communications*, 242:95 – 103, 2019.

[56] Alireza Khorshidi and Andrew A. Peterson. Amp: A modular approach to machine learning in atomistic simulations. *Computer Physics Communications*, 207:310 – 324, 2016.

[57] Nongnuch Artrith, Tobias Morawietz, and Jörg Behler. High-dimensional neural-network potentials for multicomponent systems: Applications to zinc oxide. *Phys. Rev. B*, 83:153101, Apr 2011.

[58] Nongnuch Artrith and Jörg Behler. High-dimensional neural network potentials for metal surfaces: A prototype study for copper. *Phys. Rev. B*, 85:045439, Jan 2012.

[59] Nongnuch Artrith, Björn Hiller, and Jörg Behler. Neural network potentials for metals and oxides – first applications to copper clusters at zinc oxide. *physica status solidi (b)*, 250(6):1191–1203, 2013.

[60] Stefan Chmiela, Huziel E. Sauceda, Klaus-Robert Müller, and Alexandre Tkatchenko. Towards exact molecular dynamics simulations with machine-learned force fields. *Nature Communications*, 9(1):3887, 2018.

[61] Christoph Schran, Jörg Behler, and Dominik Marx. Automated fitting of neural network potentials at coupled cluster accuracy: Protonated water clusters as testing ground. *Journal of chemical theory and computation*, 16(1):88–99, 2020.

[62] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O. Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):140022–140022, 2014.

[63] O. Anatole von Lilienfeld, Klaus-Robert Müller, and Alexandre Tkatchenko. Exploring chemical compound space with quantum-based machine learning. 2019.

[64] Kevin E Riley, Bryan T Op't Holt, and Kenneth M Merz. Critical assessment of the performance of density functional methods for several atomic and molecular properties. *Journal of chemical theory and computation*, 3(2):407–433, 2007.

[65] B. Hammer, L. B. Hansen, and J. K. Nørskov. Improved adsorption energetics within density-functional theory using revised perdew-burke-ernzerhof functionals. *Phys. Rev. B*, 59:7413–7421, Mar 1999.

Supplementary Information for "A Universal Framework for Featurization of Atomistic Systems"

0.1 Code and Saved Models

Please checkout and install the "MCSH_paper1" branch of $AMPTorch$ to try any of the saved models and test scripts in this study. The code can be found here: https://github.com/medford-group/amptorch. The test scripts can be found here: https://github.com/medford-group/GMP_AmpTorch_Tests. Tutorials for regenerating all the test models are all included in the repo.

0.2 Sample Calculation and Proof of Rotation Invariance

To show that the norms of the groups are rotation invariance, it is suffice to show that the results depend only on distances.

0.2.1 General Expression

Substitute in the definition of S_{abc} and the Gaussian functions:

$$\begin{aligned}\mu_{i,abc} &=< probe, \hat{\rho} >=< angular\ probe \times radial\ probe, \hat{\rho} > \\ &= \sum_{\substack{j \\ atoms}} \sum_{\substack{k \\ Gaussians}} \iiint_V S_{abc} G_{probe,i} G_{dens,j,k} dV \\ &= \sum_{\substack{j \\ atoms}} \sum_{\substack{k \\ Gaussians}} \iiint_V \sum_{\substack{t \\ terms}} C_t x^{m_{x,t}} y^{m_{y,t}} z^{m_{z,t}} A_i e^{-\alpha_i r^2} B_{j,k} e^{-\beta_{j,k}(r-r_{0;j})^2} dV\end{aligned} \tag{7}$$

where A, B, α, β are parameters for the Gaussian functions, and r_0's are the relative coordinates of the nearby atoms of interest.

Since the product of Gaussian functions is another Gaussian function, the equation can be written as:

$$\begin{aligned}\mu_{i,abc} &= \sum_j \sum_k \sum_t \iiint_V C_t x^{m_{x,t}} y^{m_{y,t}} z^{m_{z,t}} D_{i,j,k} e^{-\gamma_{i,j,k}(r-r'_{0;i,j,k})^2} dV \\ &= \sum_{j,k,t} K_{i,j,k,t} \int_{-\infty}^{\infty} x^{m_{x,t}} e^{-\gamma_{i,j,k}(x-x'_{0;i,j,k})^2} dx \int_{-\infty}^{\infty} y^{m_{y,t}} e^{-\gamma_{i,j,k}(y-y'_{0;i,j,k})^2} dy \int_{-\infty}^{\infty} z^{m_{z,t}} e^{-\gamma_{i,j,k}(z-z'_{0;i,j,k})^2} dz\end{aligned} \tag{8}$$

where D, K are constants, r'_0's are the center of the resulting Gaussian functions, and x'_0, y'_0, z'_0 are the components of r'_0. Since the integral

$$\int_{-\infty}^{\infty} x^m e^{-\gamma(x-x'_0)^2} dx \tag{9}$$

has analytical solutions, $\mu_{i,abc}$ is straightforward to calculate. More specifically,

$$\begin{aligned}&\int_{-\infty}^{\infty} x^0 e^{-\gamma(x-x'_0)^2} dx = \sqrt{\frac{\pi}{\gamma}} \\ &\int_{-\infty}^{\infty} x^1 e^{-\gamma(x-x'_0)^2} dx = x'_0 \sqrt{\frac{\pi}{\gamma}} \\ &\int_{-\infty}^{\infty} x^2 e^{-\gamma(x-x'_0)^2} dx = (\frac{1}{2\gamma} + x_0^{'2}) \sqrt{\frac{\pi}{\gamma}} \\ &...\end{aligned} \tag{10}$$

0.3 Rotation Invariance

To show the features $\mathfrak{p}_{i,abc}$ are rotation invariance, it is suffice to show that the values of them only depend on distances. Although a general proof that it works for all MCSH functions (S^n_{abc}) is more involved and beyond the scope of this work, it is straight forward to show few specific cases as examples, and the rest can be proved the same way.

0.3.1 $\boldsymbol{\varphi}_{i,000}$

Follow Equation 8, and note that $S_{000} = 1$ only has one term

$$\begin{aligned}\varphi_{i,000} &= \mu_{i,000} \\ &= \sum_{j,k} K_{i,j,k} \int_{-\infty}^{\infty} e^{-\gamma_{i,j,k}(x-x'_{0;i,j,k})^2} dx \int_{-\infty}^{\infty} e^{-\gamma_{i,j,k}(y-y'_{0;i,j,k})^2} dy \int_{-\infty}^{\infty} e^{-\gamma_{i,j,k}(z-z'_{0;i,j,k})^2} dz \\ &= \sum_{j,k} K_{i,j,k} (\frac{\pi}{\gamma_{i,j,k}})^{\frac{3}{2}}\end{aligned} \tag{11}$$

It is evident that this feature is rotation invariant

0.3.2 $\boldsymbol{\varphi}_{i,100}$

Follow Equation 8

$$\begin{aligned}\varphi_{i,100} &= \sqrt{\mu_{i,100}^2 + \mu_{i,010}^2 + \mu_{i,001}^2} \\ S_{100} &= x \\ S_{010} &= y \\ S_{001} &= z\end{aligned} \tag{12}$$

where

$$\begin{aligned}\mu_{i,100} &= \sum_{j,k} K_{i,j,k} \int_{-\infty}^{\infty} x e^{-\gamma_{i,j,k}(x-x'_{0;i,j,k})^2} dx \int_{-\infty}^{\infty} e^{-\gamma_{i,j,k}(y-y'_{0;i,j,k})^2} dy \int_{-\infty}^{\infty} e^{-\gamma_{i,j,k}(z-z'_{0;i,j,k})^2} dz \\ &= \sum_{j,k} K_{i,j,k} x'_{0;i,j,k} (\frac{\pi}{\gamma_{i,j,k}})^{\frac{3}{2}} \\ &= \sum_{l=1} K_{i,l} x'_{0;i,l} (\frac{\pi}{\gamma_{i,l}})^{\frac{3}{2}} \\ \mu_{i,100}^2 &= K_{i,1}^2 x_{0;i,1}^{'2} + K_{i,1} K_{i,2} x'_{0;i,l} x'_{0;i,2} + ...\end{aligned} \tag{13}$$

where l iterates through j, k. similarly,

$$\begin{aligned}\mu_{i,010}^2 &= K_{i,1}^2 y_{0;i,1}^{'2} + K_{i,1} K_{i,2} y'_{0;i,l} y'_{0;i,2} + ... \\ \mu_{i,001}^2 &= K_{i,1}^2 z_{0;i,1}^{'2} + K_{i,1} K_{i,2} z'_{0;i,l} z'_{0;i,2} + ...\end{aligned} \tag{14}$$

Threfore,

$$\begin{aligned}\varphi_{i,100} &= \sqrt{\mu_{i,100}^2 + \mu_{i,010}^2 + \mu_{i,001}^2} \\ &= \sqrt{K_{i,1}^2 (x_{0;i,1}^{'2} + y_{0;i,1}^{'2} + z_{0;i,1}^{'2}) + K_{i,1} K_{i,2} (x'_{0;i,l} x'_{0;i,2} + y'_{0;i,l} y'_{0;i,2} + z'_{0;i,l} z'_{0;i,2}) + ...} \\ &= \sqrt{K_{i,1}^2 < r'_{0;i,l}, r'_{0;i,l} > + K_{i,1} K_{i,2} < r'_{0;i,l}, r'_{0;i,2} > + ...}\end{aligned} \tag{15}$$

which is only a function of distances. Hence it is rotation invariant.

0.4 MD17 Aspirin Examples

0.4.1 Standard Training Procedure

Cutoffs for both the BP scheme and the GMP scheme are set to 10 Å. The neural networks are all trained using the same procedure: $lr = 1e^{-3}$ for 6000 epochs. For the BP vs. GMP comparison example on aspirin MD data, the batch size is chosen to be 256 images. For the force training example, the batch size is chosen to be 32 images.

0.4.2 Behler-Parrinello + HDNN Comparison Test Setups

12 sets of Behler-Parrinello feature are selected, as listed below

Set	G2		G4			$N_{feature}{}^{a}$
	η	R_s	η	ζ	γ	
1	[0.05, 0.0965, 0.1864, 0.3598, 0.6947, 1.3413 , 2.5897, 5.0]	[0, 1.5]	[0.001, 0.01, 0.03]	[1.0, 2.0, 4.0]	[1.0, -1.0]	156
2	[0.05, 0.0965, 0.1864, 0.3598, 0.6947, 1.3413 , 2.5897, 5.0]	[0, 1.5]	[0.01, 0.03]	[1.0, 4.0]	[1.0, -1.0]	96
3	[0.05, 0.0965, 0.1864, 0.3598, 0.6947, 1.3413 , 2.5897, 5.0]	[0]	[0.01]	[1.0, 4.0]	[1.0, -1.0]	48
4	[0.05, 0.0965, 0.1864, 0.3598, 0.6947, 1.3413 , 2.5897, 5.0]	[0]	[0.001, 0.01, 0.03]	[1.0, 2.0, 4.0]	[1.0, -1.0]	132
5	[0.05, 0.0965, 0.1864, 0.3598, 0.6947, 1.3413 , 2.5897, 5.0]	[0]	[0.01, 0.03]	[1.0, 4.0]	[1.0, -1.0]	72
6	[0.05, 0.0965, 0.1864, 0.3598, 0.6947, 1.3413 , 2.5897, 5.0]	[0, 1.5]	[0.01]	[1.0, 4.0]	[1.0, -1.0]	72
7	[0.05, 0.2324, 1.0772, 5.]	[0, 1.5]	[0.001, 0.01, 0.03]	[1.0, 2.0, 4.0]	[1.0, -1.0]	132
8	[0.05, 0.2324, 1.0772, 5.]	[0, 1.5]	[0.01, 0.03]	[1.0, 4.0]	[1.0, -1.0]	72
9	[0.05, 0.2324, 1.0772, 5.]	[0]	[0.01]	[1.0, 4.0]	[1.0, -1.0]	32
10	[0.05, 0.2324, 1.0772, 5.]	[0]	[0.001, 0.01, 0.03]	[1.0, 2.0, 4.0]	[1.0, -1.0]	120
11	[0.05, 0.2324, 1.0772, 5.]	[0]	[0.01, 0.03]	[1.0, 4.0]	[1.0, -1.0]	60
12	[0.05, 0.2324, 1.0772, 5.]	[0, 1.5]	[0.01]	[1.0, 4.0]	[1.0, -1.0]	48

Table 2: List of tested Behler-Parrinello feature sets. η and R_s for G2 functions are used combinatorially, same as η, ζ and γ for G4 functions. Moreover, there are 3 types of elements (C, H, O) for this dataset. Therefore, feature set 1 has $3(elements) \times 8(\eta) \times 2(R_s) + 6(possible\, element\, pairs) \times 3(\eta) \times 3(\zeta)s \times 2(\gamma) = 156$ features per atom. [a]Number of features per atom.

The list of test results with BP + HDNN models are shown below in Table 3

Set	$N_{feature}{}^a$	MAE train (meV)	MAE test (meV)	Time (ms/image)
1	156	73.0 ± 3.0	98.8 ± 2.7	22.1 ± 0.5
2	96	74.2 ± 1.4	98.8 ± 1.7	13.3 ± 0.4
3	48	71.8 ± 1.9	97.3 ± 1.8	8.2 ± 0.1
4	132	73.5 ± 1.6	100.3 ± 2.5	21.2 ± 0.3
5	72	73.2 ± 1.2	99.6 ± 1.5	12.1 ± 0.1
6	72	74.4 ± 1.6	97.4 ± 2.2	9.1 ± 0.4
7	132	73.3 ± 1.4	99.9 ± 1.5	20.9 ± 0.1
8	72	71.6 ± 1.5	97.8 ± 2.4	12.1 ± 0.1
9	32	70.5 ± 2.2	97.3 ± 1.9	8.1 ± 0.2
10	120	73.5 ± 1.5	101.2 ± 1.4	20.4 ± 0.0
11	60	71.7 ± 2.1	99.3 ± 1.6	11.8 ± 0.1
12	48	72.4 ± 1.7	96.9 ± 1.3	8.2 ± 0.1

Table 3: Performance test results of the tested GMP + HDNN setups. The values are the average values of the 10 trials, and the uncertainties are estimated by their standard deviation. ^{a}Number of features per atom.

0.4.3 GMP+SNN Comparison Test Setups

The probe of GMP has two parts: groups of MCSHs for probing angular features and radial gaussian for probing radial features. In this work, all orders of MCSH up to the indicated order are included (starting from order 0), and the number of possible groups for each order are listed below in Table **??**:

We combine the radial probes with the lists of radial Gaussians combinatorially to obtain the full list of probes/features. The list of widths (standard deviations) of the probe Gaussians is simply chosen to be uniform spaced values up to 2.0 (in the language of python, it is: linspace(0, 2.0, n_gaussian+1, endpoint=True)[1:])

Therefore, when there are 5 Gaussians with MCSH up to order 6, there are $5 \times 7 = 35$ descriptors per atom. The complete list of test results with GMP+SNN is give in Table 4.

Num. Gaussians[a]	MCSH order[b]	$N_{feature}$[c]	MAE train (meV)	MAE test (meV)	Time (ms/image)
2	0	2	164.9 ± 0.7	168.3 ± 2.0	2.9 ± 0.3
2	1	4	117.3 ± 1.1	129.1 ± 1.3	2.9 ± 0.1
2	2	6	94.8 ± 1.2	110.5 ± 0.8	3.2 ± 0.2
2	3	8	84.2 ± 0.8	102.7 ± 1.7	3.8 ± 0.2
2	4	10	79.2 ± 1.0	99.3 ± 1.3	4.5 ± 0.0
2	5	12	76.0 ± 2.3	95.4 ± 3.2	5.8 ± 0.0
2	6	14	74.2 ± 1.7	95.0 ± 2.6	8.0 ± 0.3
2	7	16	71.3 ± 1.0	91.4 ± 1.3	11.2 ± 0.3
2	8	18	72.5 ± 0.7	93.7 ± 0.6	14.9 ± 0.3
4	0	4	117.7 ± 0.6	122.1 ± 1.4	2.9 ± 0.1
4	1	8	68.2 ± 1.5	72.9 ± 1.4	3.2 ± 0.1
4	2	12	40.6 ± 0.6	44.6 ± 0.8	3.6 ± 0.0
4	3	16	33.2 ± 0.6	37.3 ± 0.5	4.7 ± 0.1
4	4	20	30.3 ± 0.7	34.7 ± 0.7	6.4 ± 0.1
4	5	24	28.2 ± 0.4	32.6 ± 0.6	8.9 ± 0.0
4	6	28	26.1 ± 0.5	30.1 ± 0.5	12.9 ± 0.2
4	7	32	24.3 ± 0.3	28.5 ± 0.4	19.3 ± 0.0
4	8	36	23.5 ± 0.2	27.5 ± 0.5	27.2 ± 0.2
6	0	6	76.3 ± 1.6	78.5 ± 1.7	3.0 ± 0.2
6	1	12	50.4 ± 0.4	54.2 ± 0.7	3.4 ± 0.2
6	2	18	35.2 ± 0.7	38.7 ± 0.7	4.1 ± 0.1
6	3	24	25.5 ± 0.3	28.7 ± 0.2	5.7 ± 0.1
6	4	30	24.1 ± 0.4	26.9 ± 0.4	8.3 ± 0.2
6	5	36	22.1 ± 0.4	25.5 ± 0.3	12.2 ± 0.3
6	6	42	20.4 ± 0.2	23.6 ± 0.1	18.2 ± 0.7
6	7	48	19.8 ± 0.6	23.1 ± 0.4	27.8 ± 0.3
6	8	54	19.1 ± 0.4	22.5 ± 0.5	39.1 ± 0.2
8	0	8	66.9 ± 0.9	68.2 ± 1.0	3.3 ± 0.2
8	1	16	48.3 ± 1.0	51.8 ± 0.8	3.7 ± 0.3
8	2	24	32.0 ± 0.2	34.8 ± 0.5	4.9 ± 0.4
8	3	32	25.7 ± 0.3	28.6 ± 0.4	7.1 ± 0.4
8	4	40	23.0 ± 0.3	25.8 ± 0.4	10.1 ± 0.4
8	5	48	21.0 ± 0.5	24.2 ± 0.4	15.5 ± 0.6
8	6	56	20.0 ± 0.3	22.8 ± 0.1	23.1 ± 0.6
8	7	64	19.3 ± 0.2	22.3 ± 0.2	36.1 ± 1.0
8	8	72	18.0 ± 0.4	21.0 ± 0.4	51.1 ± 0.8
10	0	10	66.3 ± 1.0	67.7 ± 0.9	3.2 ± 0.1
10	1	20	48.3 ± 0.2	51.8 ± 0.4	3.7 ± 0.0
10	2	30	32.4 ± 0.6	35.3 ± 0.7	5.0 ± 0.2
10	3	40	24.4 ± 0.4	27.2 ± 0.4	7.8 ± 0.1
10	4	50	22.4 ± 0.5	25.2 ± 0.5	11.8 ± 0.1
10	5	60	20.6 ± 0.3	23.4 ± 0.3	18.4 ± 0.1
10	6	70	18.8 ± 0.2	21.7 ± 0.3	28.2 ± 0.3
10	7	80	18.1 ± 0.3	20.9 ± 0.3	44.0 ± 0.6
10	8	90	16.8 ± 0.2	19.7 ± 0.2	63.9 ± 1.3
12	0	12	65.9 ± 1.0	67.5 ± 1.2	3.4 ± 0.2
12	1	24	47.8 ± 0.7	51.2 ± 0.4	3.9 ± 0.0
12	2	36	32.3 ± 1.1	35.0 ± 0.7	5.5 ± 0.2
12	3	48	24.4 ± 0.6	27.2 ± 0.6	8.8 ± 0.2
12	4	60	22.2 ± 0.3	25.1 ± 0.4	14.2 ± 0.8
12	5	72	20.3 ± 0.3	22.9 ± 0.4	21.8 ± 0.8
12	6	84	18.6 ± 0.4	21.3 ± 0.3	33.6 ± 0.8
12	7	96	17.7 ± 0.3	20.5 ± 0.3	52.0 ± 0.6
12	8	108	17.0 ± 0.3	19.7 ± 0.1	74.9 ± 1.9
14	0	14	66.2 ± 1.6	67.8 ± 1.8	3.6 ± 0.3
14	1	28	48.7 ± 0.7	51.7 ± 0.8	4.2 ± 0.1
14	2	42	32.3 ± 0.5	35.2 ± 0.6	6.0 ± 0.1
14	3	56	24.2 ± 0.3	27.0 ± 0.3	9.7 ± 0.1
14	4	70	22.6 ± 0.2	25.4 ± 0.3	15.7 ± 0.7
14	5	84	20.3 ± 0.2	23.2 ± 0.3	24.5 ± 0.1
14	6	98	18.8 ± 0.1	21.5 ± 0.1	38.1 ± 0.3
14	7	112	17.8 ± 0.3	20.7 ± 0.2	60.1 ± 0.7
14	8	126	16.8 ± 0.3	19.6 ± 0.2	88.6 ± 3.5
16	0	16	66.5 ± 1.3	67.7 ± 1.2	3.5 ± 0.1
16	1	32	47.7 ± 0.4	51.0 ± 0.4	4.3 ± 0.0
16	2	48	32.1 ± 0.3	34.7 ± 0.4	6.4 ± 0.1
16	3	64	24.5 ± 0.3	27.2 ± 0.1	10.7 ± 0.1
16	4	80	22.3 ± 0.5	25.0 ± 0.6	17.7 ± 0.8
16	5	96	20.0 ± 0.3	22.8 ± 0.3	28.4 ± 1.0
16	6	112	18.6 ± 0.1	21.2 ± 0.2	43.7 ± 1.1
16	7	128	17.9 ± 0.2	20.6 ± 0.4	69.5 ± 2.0
16	8	144	16.9 ± 0.2	19.7 ± 0.3	101.5 ± 5.0
18	0	18	65.6 ± 1.2	67.1 ± 1.2	3.6 ± 0.0
18	1	36	47.7 ± 0.4	51.0 ± 0.4	4.7 ± 0.1
18	2	54	32.0 ± 0.4	34.5 ± 0.6	6.8 ± 0.0
18	3	72	24.6 ± 0.3	27.4 ± 0.2	11.7 ± 0.1
18	4	90	22.0 ± 0.2	24.8 ± 0.3	18.9 ± 0.0
18	5	108	20.2 ± 0.2	22.9 ± 0.3	30.7 ± 0.1
18	6	126	18.4 ± 0.3	21.1 ± 0.2	47.9 ± 0.0
18	7	144	17.5 ± 0.4	20.3 ± 0.3	75.8 ± 0.2
18	8	162	16.7 ± 0.1	19.5 ± 0.2	109.2 ± 0.1
20	0	20	65.3 ± 0.4	67.0 ± 0.6	3.7 ± 0.0
20	1	40	48.7 ± 0.5	51.7 ± 0.4	4.9 ± 0.2
20	2	60	31.9 ± 0.3	34.9 ± 0.4	7.3 ± 0.1
20	3	80	24.7 ± 0.3	27.4 ± 0.1	12.8 ± 0.4
20	4	100	22.2 ± 0.5	25.0 ± 0.5	20.9 ± 0.3
20	5	120	20.4 ± 0.2	23.3 ± 0.2	33.9 ± 0.1
20	6	140	18.6 ± 0.4	21.3 ± 0.4	53.0 ± 0.1
20	7	160	17.8 ± 0.4	20.6 ± 0.4	83.7 ± 0.1
20	8	180	16.9 ± 0.2	19.6 ± 0.2	121.3 ± 1.0

Table 4: Performance test results of the full sets of tested GMP + SNN setups. The values are the average values of the 10 trials, and the uncertainties are estimated by their standard deviation. [a]Number of possible Gaussian functions used to construct the descriptor probes. [b]The highest MCSH order used to construct the probes. For example, when highest order is 2, that means all groups from MCSH of order 0, 1 and 2 are used to construct the probes. [c]Number of features per atom.

Model	Sigmas	$N_{feature}$
$GMP(6,3) + SNN(50,3)$	[0.333, 0.666, 1.0, 1.333, 1.666, 2.0]	24
$GMP(10,6) + SNN(50,3)$	[0.2,0.4,0.6,0.8,1.0,1.2,1.4,1.6,1.8,2.0]	70

Table 5: Setups for the GMP+SNN models used in the force training example

0.4.4 Force Training Example

The sigmas of the radial probe Gaussians are listed in Table 5

0.5 QM9 Example

0.5.1 Per-element Bias

A per-element bias is added to the SNN model to improve performance. Conceptually, this is equivalent to fitting to formation energies rather than absolute energies. The total energy of an atom is the model predicted energy plus the per-element bias of the specific atom type. To determine the bias, a linear model is applied. The number of atoms for each element types are counted for all the images in the training set, and they are the independent variable. The corresponding energies for each system is the dependent variable. For example, the per-element bias of the trials with 100K training images is shown below in Table 6:

Atom Type	Per-element Bias (meV)
H	-2795.2721
C	-6217.7719
N	-4552.4431
O	-4432.6761
F	-4075.4931

Table 6: Per-element Bias of each atom type found by the linear model, for the 100K training set

0.5.2 Training Procedure

With the per-element bias determined, GMP+SNN models are fitted to the atomization energy minus the per-element biases. The model setups are given in Table 7. The cutoff distance is always 15 Å, so that the largest radial probe takes a negligible value of $8 \times 10^{-}4$ at the cutoff. The models are trained for 12,000 epochs with learning rate decrease by factor of 2 every 2,000 epochs, from 1^{-2} to 3^{-4}. The batch size is set to be 32 images.

0.5.3 Transfer Learning to New Element

For this example, we used the same procedure as above, with the caveat that the per-element biases are not fitted using a linear model, but directly pulled from the 100k molecule trial. For more detail please refer to the test scripts.

Shown in Figure 6 is the error distribution of the system containing F atoms in the basis of error per F atom. Shown in Figure 7 are the error distributions from different trials with different random seeds.

Model	Sigmas	$N_{feature}$	$N_{parameters}$
GMP(10,1)+SNN(32,32,32)	linspace(0.02, 2.0, 10, endpoint=True)	20	3009
GMP(30,2)+SNN(128,64,64)	linspace(0.02, 2.0, 30, endpoint=True)	90	24641
GMP(50,4)+SNN(256,128,64)	linspace(0.02, 2.0, 50, endpoint=True)	250	106369
GMP(70,6)+SNN(512,256,128,64)	linspace(0.02, 2.0, 70, endpoint=True)	490	425857

Table 7: Setups for the GMP+SNN models used in the QM9 examples. Cutoff distance is always 15 Å. $N_{parameters}$ is the number of trainable parameters of the neural network model.

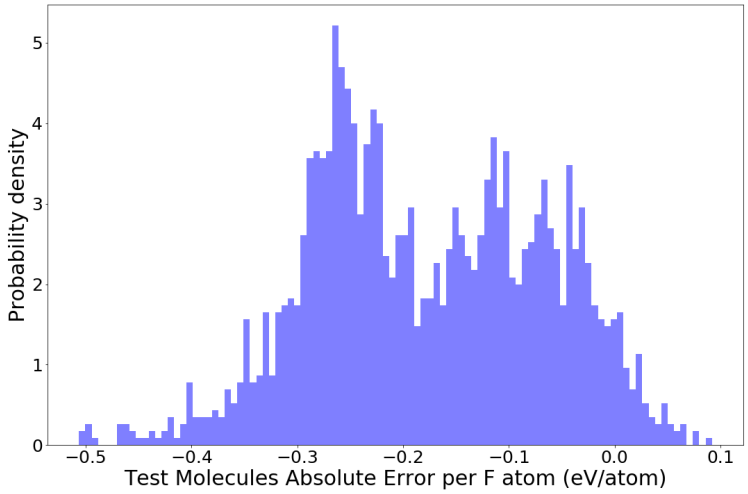


Figure 6: Error distribution of F-containing molecules per F atom.

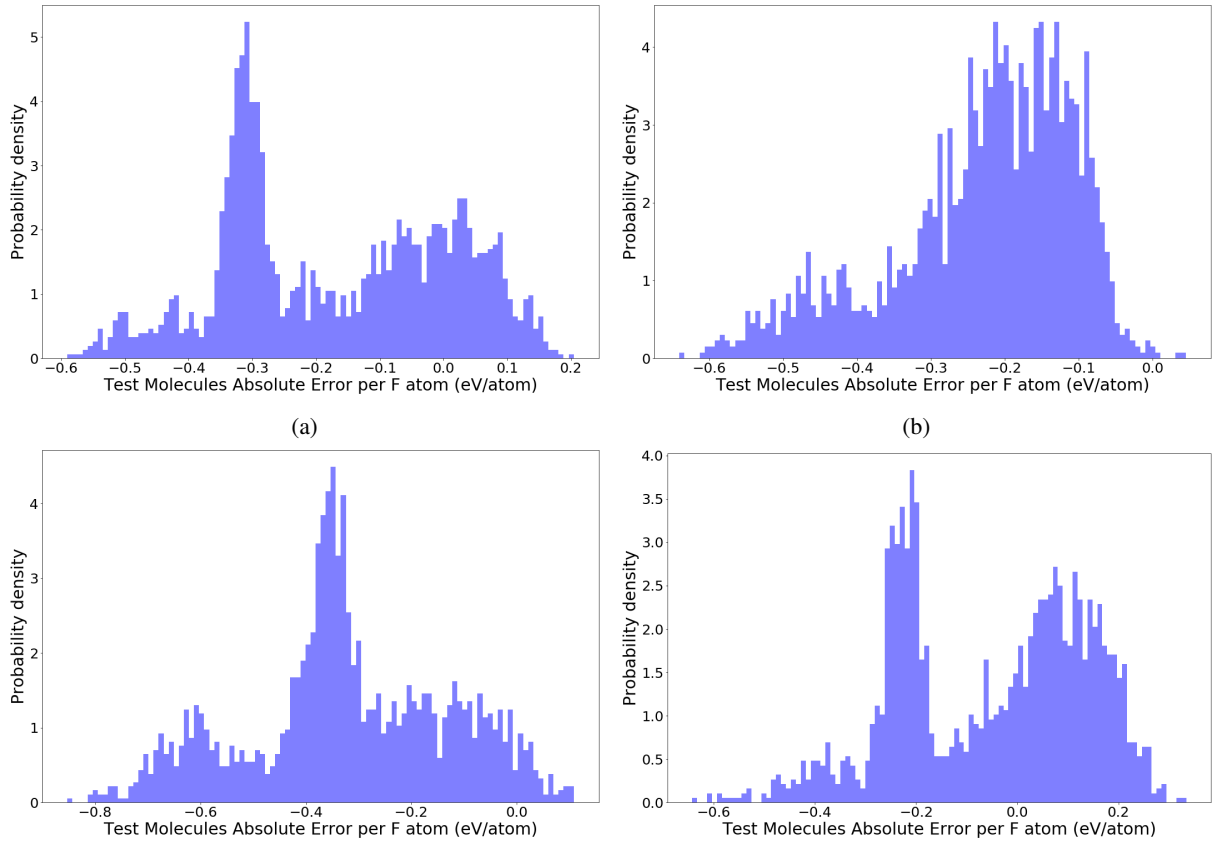


Figure 7: Error distribution of prediction for F-containing molecules with different random seed for the model trained on molecules without F.

Model	Sigmas	$N_{feature}$	$N_{parameters}$
GMP(30,2)+SNN(128,64,64)	linspace(0.02, 2.0, 30, endpoint=True)	90	24641
GMP(50,4)+SNN(256,128,64)	linspace(0.02, 2.0, 50, endpoint=True)	250	106369
GMP(70,6)+SNN(512,256,128,64)	linspace(0.02, 2.0, 70, endpoint=True)	490	425857
GMP(90,8)+SNN(1024,512,128,64)	linspace(0.02, 2.0, 90, endpoint=True)	910	1432705

Table 8: Setups for the GMP+SNN models used in the OC20 examples. Cutoff distance is always 15 Å. $N_{parameters}$ is the number of trainable parameters (weights) of the neural network model.