SE(3)-Equivariant Graph Neural Networks for Data-Efficient and Accurate Interatomic Potentials

Simon Batzner,[1] Tess E. Smidt,[2] Lixin Sun,[1] Jonathan P. Mailoa,[3]
Mordechai Kornbluth,[3] Nicola Molinari,[1] and Boris Kozinsky[1,3]

[1]*Harvard University*
[2]*Lawrence Berkeley National Laboratory*
[3]*Robert Bosch Research and Technology Center*

This work presents Neural Equivariant Interatomic Potentials (NequIP), a SE(3)-equivariant neural network approach for learning interatomic potentials from *ab-initio* calculations for molecular dynamics simulations. While most contemporary symmetry-aware models use invariant convolutions and only act on scalars, NequIP employs SE(3)-equivariant convolutions for interactions of geometric tensors, resulting in a more information-rich and faithful representation of atomic environments. The method achieves state-of-the-art accuracy on a challenging set of diverse molecules and materials while exhibiting remarkable data efficiency. NequIP outperforms existing models with up to three orders of magnitude fewer training data, challenging the widely held belief that deep neural networks require massive training sets. The high data efficiency of the method allows for the construction of accurate potentials using high-order quantum chemical level of theory as reference and enables high-fidelity molecular dynamics simulations over long time scales.

INTRODUCTION

Molecular dynamics (MD) simulations are an indispensable tool for computational discovery in fields as diverse as energy storage, catalysis, and biological processes [1–3]. While the atomic forces required to integrate Newton's equations of motion can in principle be obtained with high fidelity from quantum-mechanical calculations such as density functional theory (DFT), in practice the unfavorable computational scaling of first-principles methods limits simulations to short time scales and small numbers of atoms. This prohibits the study of many interesting physical phenomena beyond the time and length scales that are currently accessible, even on the largest supercomputers. Owing to their simple functional form, classical models for the atomic potential energy can typically be evaluated orders of magnitude faster than using first-principles methods, thereby enabling the study of large numbers of atoms over long time scales. However, due to their limited mathematical form, classical interatomic potentials, or force fields, are inherently limited in their predictive accuracy which has historically led to a fundamental trade-off between obtaining high computational efficiency while also predicting faithful dynamics of the system under study. The construction of flexible models of the interatomic potential energy based on Machine Learning (ML-IP), and in particular Neural Networks (NN-IP), has shown great promise in providing a way to move past this dilemma, promising to learn high-fidelity potentials from *ab-initio* reference calculations while retaining favorable computational efficiency [4–13]. One of the limiting factors of NN-IPs is that they typically require collection of large training sets of *ab-initio* calculations, often including thousands or even millions of reference structures [4, 9, 10, 14–16]. This computationally expensive process of training data collection has severely limited the adoption of NN-IPs as it quickly becomes a bottleneck in the development of force-fields for new systems. Kernel-based approaches, such as e.g. Gaussian Processes (GP) [5, 8] or Kernel Ridge Regression (KRR) [17], are a way to remedy this problem as they often generalize better from limited sample sizes. However, such methods generally tend to exhibit poor computational scaling with the number of reference configurations, in both training (cubic in training set size) and prediction (linear in training set size). This limits both the amount of training data they can be trained on as well as the length and size of simulations that can be simulated with them.

In this work, we present the Neural Equivariant Interatomic Potential (NequIP), a highly data-efficient deep learning approach for learning interatomic potentials from reference first-principles calculations. We show that the proposed method obtains high accuracy compared to existing ML-IP methods across a wide variety of systems, including small molecules, water in different phases, an amorphous solid, a reaction at a solid/gas interface, and a Lithium superionic conductor. Furthermore, we find that NequIP exhibits exceptional data efficiency, enabling the construction of accurate interatomic potentials from limited data sets of fewer than 1,000 or even as little as 100 reference *ab-initio* calculations, where other methods require orders of magnitude more. It is worth noting that on small molecular data sets, NequIP outperforms not only other neural networks, but is also competitive with kernel-based approaches, which typically obtain better predictive accuracy than NN-IPs on small data sets (although at significant additional cost scaling in training and prediction). We further demonstrate high data

efficiency and accuracy with state-of-the-art results on a training set of molecular data obtained at the quantum chemical coupled-cluster level of theory. Finally, we validate the method through a series of simulations and demonstrate that we can reproduce with high fidelity structural and kinetic properties computed from NequIP simulations in comparison to *ab-initio* molecular dynamics simulations (AIMD). We directly verify that the performance gains are connected with the unique SE(3)-equivariant convolution architecture of the new NequIP model.

Related Work

First applications of machine learning for the development of interatomic potentials were built on descriptor-based approaches combined with shallow neural networks or Gaussian Processes [4, 5], designed to exhibit invariance with respect to translation, permutation of atoms of the same chemical species, and rotation. Recently, rotationally invariant graph neural networks (GNN-IPs) have emerged as a powerful architecture for deep learning of interatomic potentials that eliminates the need for hand-crafted descriptors and allows to instead learn representations on graphs of atoms from invariant features of geometric data (e.g. radial distances or angles) [9–11, 13]. In GNN-IPs, atomic structures are represented by collections of nodes and edges, where nodes in the graph correspond to individual atoms and edges are typically defined by simply connecting every atom to all other atoms that are closer than some cutoff distance r_c. Every node/atom i is associated with a feature $\mathbf{h}_i \in \mathbb{R}^h$, consisting of scalar values, which is iteratively refined via a series of convolutions over neighboring atoms j based on both the distance to neighboring atoms r_{ij} and their features $\mathbf{h}_j$. This iterative process allows information to be propagated along the atomic graph through a series of convolutional layers and can be viewed as a message-passing scheme [18]. Operating only on interatomic distances allows GNN-IPs to be rotation- and translation-invariant, making both the output as well as features internal to the network invariant to rotations. In contrast, the method outlined in this work uses relative position *vectors* rather than simply distances (scalars), which makes internal features instead *equivariant* to rotation and allows for angular information to be used by rotationally equivariant filters. Similar to other methods, we can restrict convolutions to only a local subset of all other atoms that lie closer to the central atom than a chosen cutoff distance r_c, see Figure 1, left.

A series of related methods have recently been proposed: DimeNet [11] expands on using pairwise interactions in a single convolution to include angular, three-body terms, but individual features are still comprised of scalars (distances and three-body angles are invariant to rotation), as opposed to vectors used in this work. Another central difference to NequIP is that DimeNet explicitly enumerates angles between pairs of atoms and operates on a basis embedding of distances and angles, whereas NequIP operates on relative position *vectors* and a basis embedding of distances, and thus never explicitly computes three-body angles. Cormorant [19] uses an equivariant neural network for property prediction on small molecules. This method is demonstrated on potential energies of small molecules but not on atomic forces or systems with periodic boundary conditions. Townshend et al. [20] use the framework of Tensor-Field Networks [21] to directly predict atomic force vectors. The predicted forces are not guaranteed by construction to conserve energy since they are not obtained as gradients of the total potential energy. This may lead to problems in simulations of molecular dynamics over long times. None of these three works [11, 19, 20] demonstrates capability to perform molecular dynamics simulations.

In this work we present a deep learning energy-conserving interatomic potential for both molecules and materials built on SE(3)-equivariant convolutions over geometric tensors that yields state-of-the-art accuracy, outstanding data-efficiency, and can with high fidelity reproduce structural and kinetic properties from molecular dynamics simulations.

RESULTS

Equivariance

The concept of equivariance arises naturally in machine learning of atomistic systems (see e.g. [22]): physical properties have well-defined transformation properties under translation and rotation of a set of atoms. As a simple example, if a molecule is rotated in space, the vectors of its atomic dipoles or forces also rotate accordingly, via equivariant transformation. Equivariant neural networks are able to more generally represent tensor properties and tensor operations of physical systems (e.g. vector addition, dot products, and cross products). Equivariant neural networks are guaranteed to preserve the known transformation properties of physical systems under a change of coordinates because they are explicitly constructed from equivariant operations. Formally, a function $f : X \to Y$ is equivariant with respect to a group G that acts on X and Y if:

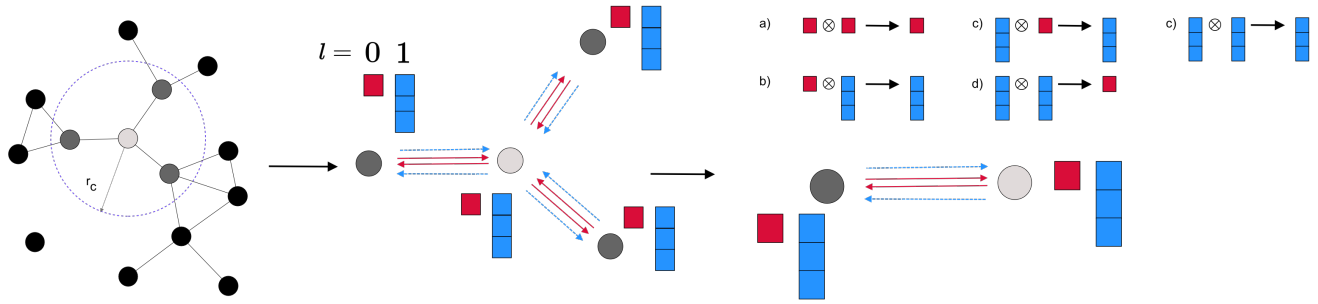

FIG. 1: Left: a set of atoms is interpreted as an atomic graph with local neighborhoods. Middle: every atom carries a set of scalar and vector features with it. Right: atoms exchange information via filters, that are again scalars and vectors. The interactions of features and filters define five interactions.

$$D_Y[g]f(x) = f(D_X[g]x) \quad \forall g \in G, \forall x \in X \tag{1}$$

where $D_X[g]$ and $D_Y[g]$ are the representations of the group element g in the vector spaces X and Y, respectively. In this work, we focus on equivariance with respect to SE(3), i.e. the group of rotations and translations in 3D space.

Neural Equivariant Interatomic Potentials

Given a set of atoms (a molecule or a material), we aim to find a mapping from atomic positions $\vec{r}_i$ and chemical species (identified by atomic numbers Z_i) to the total potential energy and the forces acting on the atoms:

$$f : \{\vec{r}_i, Z_i\} \rightarrow E_{pot} \tag{2}$$

Forces are obtained as gradients of the predicted potential energy with respect to the atomic positions, which guarantees energy conservation:

$$\vec{F}_i = -\nabla_i E_{pot} \tag{3}$$

Gradients can be obtained with relatively low computational overhead in modern auto-differentiation frameworks such as TensorFlow or PyTorch [23, 24]. Following previous work [4], we further define the total potential energy of the system as a sum of atomic potential energies:

$$E_{pot} = \sum_{i \in N_{atoms}} E_{i,atomic} \tag{4}$$

These atomic local energies $E_{i,atomic}$ are the scalar node attributes predicted by the graph neural network. Even though the output of NequIP is the predicted potential energy E_{pot}, which is invariant under translations and rotations, the network contains *internal features* that are tensors which are equivariant to rotation. This constitutes the core difference between NequIP and existing scalar-valued invariant GNN-IPs. The remainder of this section will discuss the design of the network in further detail.

A series of methods has been introduced to realize rotationally equivariant neural networks [13, 21, 25, 26]. Here, we build on the layers introduced in Tensor-Field Networks (TFN) [21], which enable the construction of neural networks that exhibit equivariance to translation, permutation, and rotation. Every atom in NequIP is associated with a feature comprised of tensors of different order: scalars, vectors, and higher-order tensors. Formally, these features are geometric objects that comprise a direct sum of irreducible representations of the SO(3) symmetry group. Second, the convolutions that operate on these geometric objects are equivariant functions instead of invariant ones, i.e. if a feature at layer k is rotated, then the output of the convolution from layer $k \rightarrow k+1$ rotates accordingly. In practice, the features are implemented as a dictionary $V^{(l)}_{acm}$ with keys l, where $l = 0, 1, 2, ...$ is a non-negative integer and is called the "rotation order", labeling the irreducible representations. The indices a, c, m, correspond to the atoms, the channels (elements of the feature), and the representation index which takes values $m \in [-l, l]$, respectively.

Convolution operations are naturally translation invariant, since their filters act on relative interatomic distance vectors. Moreover, they are permutation invariant since all convolution contributions are summed. Note that while atomic features are equivariant to permutation of atom indices, globally, the total potential energy of the system is invariant to permutation. To achieve rotation equivariance, the convolution filters are constrained to be products of learnable radial functions and spherical harmonics, which are equivariant under

SO(3) [21]:

$$F(\vec{r}_{ij}) = R(r_{ij})Y_m^{(l)}(\hat{r}_{ij}) \tag{5}$$

where if $\vec{r}_{ij}$ denotes the relative position from central atom i to neighboring atom j, $\hat{r}_{ij}$ and r_{ij} are the associated unit vector and interatomic distance, respectively. It should be noted that all learnable weights in the filter lie in the rotationally invariant radial function $R(r_{ij})$. This radial function is implemented as a small neural network with one hidden layer and a shifted softplus activation function [9], operating on interatomic distances expressed in a basis of choice, $R(r_{ij}) : \mathbb{R}^{N_b} \to \mathbb{R}^h$, where N_b is the number of basis functions and h is the feature dimension:

$$R(r_{ij}) = W_2 \ln(0.5 \exp(W_1 B(r_{ij})) + 0.5) \tag{6}$$

where $W_1 \in \mathbb{R}^{N_{hidden} \times N_b}$ and $W_2 \in \mathbb{R}^{h \times N_{hidden}}$ are weight matrices, h is the dimension of the feature and N_{hidden} is the dimension of the hidden layer in the feed-forward neural network (in our experiments, we use $N_{hidden} = N_b$, resulting in comparatively small neural networks for the radial function). Radial Bessel functions and a polynomial envelope function f_{env} discussed in recent work [11] are used to expand the interatomic distances:

$$B(r_{ij}) = \sqrt{\frac{2}{r_c}} \frac{\sin(\frac{n\pi}{r_c} r_{ij})}{r_{ij}} f_{env}(r_{ij}, r_c) \tag{7}$$

where r_c is a local cutoff radius, restricting interactions to atoms closer than some cutoff distance and f_{env} is the polynomial defined in [11] with $p = 6$ operating on the interatomic distances normalized by the cutoff radius $\frac{r_{ij}}{r_c}$. The use of cutoffs/local atomic environments allows the computational cost of evaluation to scale linearly with the number of atoms. Similar to [11], we initialize the Bessel functions with $n = [1, 2, ..., N_b]$ and subsequently optimize $n\pi$ via backpropagation rather than keeping it constant. For systems with periodic boundary conditions, we use the `neighbor_list` functionality as implemented in the ASE code [27] to identify appropriate atomic neighbors and then convolve over them.

Finally, in the convolution, the input feature tensor and the filter have to again be combined in an equivariant manner, which is achieved via a geometric tensor product, yielding an output feature that again is rotationally equivariant. A tensor product of two geometric tensors is computed via Clebsch-Gordan coefficients, as outlined in [21]. Since NequIP deals with force vectors, the network design is simplified by only using scalar (l=0) and vector (l=1) representations. Thus, we can enumerate five distinct products or "interactions" between $l = 0$ and $l = 1$ filters and $l = 0$ and $l = 1$ features that correspond to simple operations between scalars and vectors:

- $0 \otimes 0 \to 0$ (product of two scalars)
- $0 \otimes 1 \to 1$ (scalar multiplication of a vector)
- $1 \otimes 0 \to 1$ (scalar multiplication of a vector)
- $1 \otimes 1 \to 0$ (dot product of two vectors)
- $1 \otimes 1 \to 1$ (cross product of two vectors)

It is trivial to include higher-order interactions, and previous works have increased the rotation order beyond $l = 1$ [20, 28]. However, it should be noted that every interaction comes with additional trainable radial functions and hence additional weights, which thus adds to the model capacity, increasing the number of model weights and the memory footprint of the model. Omitting all higher-order interactions that go beyond the $0 \otimes 0 \to 0$ interaction will result in a conventional GNN-IP with invariant convolutions over scalar features, similar to e.g. SchNet [9]. Finally, as outlined in [21], a full convolutional layer $\mathcal{L}$ implementing an interaction with filter f acting on an input i producing output o: $l_f \otimes l_i \to l_o$ is given by:

$$\mathcal{L}_{acm_o}^{(l_o)}(\vec{r}_a, V_{acm_i}^{(l_i)}) = \sum_{m_f, m_i} C_{(l_f,m_f)(l_i,m_i)}^{(l_o,m_o)} \sum_{b \in S} R_c^{(l_f,l_i)}(r_{ab}) Y_{m_f}^{(l_f)}(\hat{r}_{ab}) V_{bcm_i}^{(l_i)} \tag{8}$$

where a and b index the central atom of the convolution and the neighboring atom $b \in S$, respectively, and C indicates the Clebsch-Gordan coefficients. To illustrate that the interactions outlined above reduce to a set of five simple operations, we write out a full $1 \otimes 1 \to 1$ interaction, i.e. a convolution that uses a $l = 1$ filter to operate on a $l = 1$ feature, yielding again a $l = 1$ output. Given the notation above, this corresponds to $l_i = l_f = l_o = 1$, facilitating a cross-product interaction between two $l = 1$ tensors. In this case, the Clebsch-Gordan coefficients reduce to the Levi-Civita symbol [21]:

$$C^{(l_0=1,m_o)}_{(l_f=1,m_f),(l_i=1,m_i)} \propto \epsilon_{ofi} = \begin{cases} 1 & (o,f,i) \in \{(1,2,3),(2,3,1),(3,1,2)\} \\ -1 & (o,f,i) \in \{(1,3,2),(2,1,3),(3,2,1)\} \\ 0 & \text{else} \end{cases} \tag{9}$$

Evaluating ϵ_{ofi} and using the relationship $Y^{(1)}(\hat{r}) \propto \hat{r}$, we recognize the output as the vector cross product, taken here between the relative positions and the input feature element $V^{(l=1)}_{bc}$:

$$\mathcal{L}^{(l_o=1)}_{ac}(\vec{r}_a, V^{(l_i=1)}_{ac}) = \begin{pmatrix} \sum_{b\in B} R_c(r_{ab})\hat{r}_2 V_{bc3} - \sum_{b\in B} R_c(r_{ab})\hat{r}_3 V_{bc2} \\ \sum_{b\in B} R_c(r_{ab})\hat{r}_3 V_{bc1} - \sum_{b\in B} R_c(r_{ab})\hat{r}_1 V_{bc3} \\ \sum_{b\in B} R_c(r_{ab})\hat{r}_1 V_{bc2} - \sum_{b\in B} R_c(r_{ab})\hat{r}_2 V_{bc1} \end{pmatrix} \tag{10}$$

After every convolution, output tensors of a rotation order l stemming from different tensor products are concatenated on a per-atom basis. To update atomic features, the model also leverages self-interaction layers similar to SchNet [9], corresponding to dense layers that are applied in an atom-wise fashion with weights shared across atoms. While different weights are used for different rotation orders, the same set of weights is applied for all representation indices m of a given rotation order l. Shifted softplus nonlinearities [9] are used as rotation-equivariant nonlinearities as introduced in [21], which are applied to the Euclidean norm of the input feature, the output of which is in turn combined with the input tensor, thus preserving overall equivariance.

The NequIP network architecture, shown in Figure 2, is built on an atomic embedding, followed by a series of interaction blocks, and finally an output block:

- **Embedding**: following SchNet, the initial feature is generated using a trainable embedding, that operates on the atomic number Z_i alone, implemented via a self-interaction layer.
- **Interaction Block**: interaction blocks encode interactions between neighboring atoms: the core of this block is the convolution function, outlined in equation 8. For every output rotation order l_o, the features from different tensor product interactions are concatenated to give a new feature, which is in return refined with atom-wise self-interaction layers and equivariant non-linearities. We equip interactions blocks with a ResNet-style update [29] where the input feature $\mathbf{x}$ is updated atom-wise via the output of an interaction block $f(\mathbf{x})$ that gives the final feature $r(\mathbf{x}) = f(\mathbf{x}) + \mathbf{x}$ (features are added element-wise in the m-dimension). Note that this operation is equivariant since the addition of an equivariant feature $\mathbf{x}$ and an equivariant function $f(\mathbf{x})$ preserves equivariance.
- **Output Block**: the $l = 0$ feature of the final convolution is passed to an output block, which consists of another atom-wise self-interaction layer, an equivariant non-linearity, and a final atom-wise self-interaction layer.

The scalar atomic outputs of the final layer can be interpreted as atomic potential energies which are summed to give the total predicted potential energy of the system (Equation 4). Forces are subsequently obtained as the negative gradient of the predicted total potential energy, thereby ensuring both energy conservation as well as rotation-equivariant forces (see equation 3).

Experiments

We validate the proposed method on a series of diverse and challenging data sets: first we demonstrate that we improve upon state-of-the-art accuracy on MD-17, a data set of small, organic molecules that is widely used for benchmarking ML-IPs [9, 11, 17, 30, 31]. Next, we show that NequIP can also accurately learn forces obtained on small molecules at the quantum chemical CCSD(T) level [31], opening the door to scalable and efficient molecular dynamics simulations with beyond-DFT accuracy. To broaden the applicability of the method beyond small isolated molecules, we explore a series of extended systems with periodic boundary conditions, consisting of both surfaces and bulk materials: water in different phases [15, 32], a chemical reaction at a solid/gas interface, an amorphous Lithium Phosphate [12], and a Li superionic conductor [13]. Details of the training procedure are provided in the Methods section.

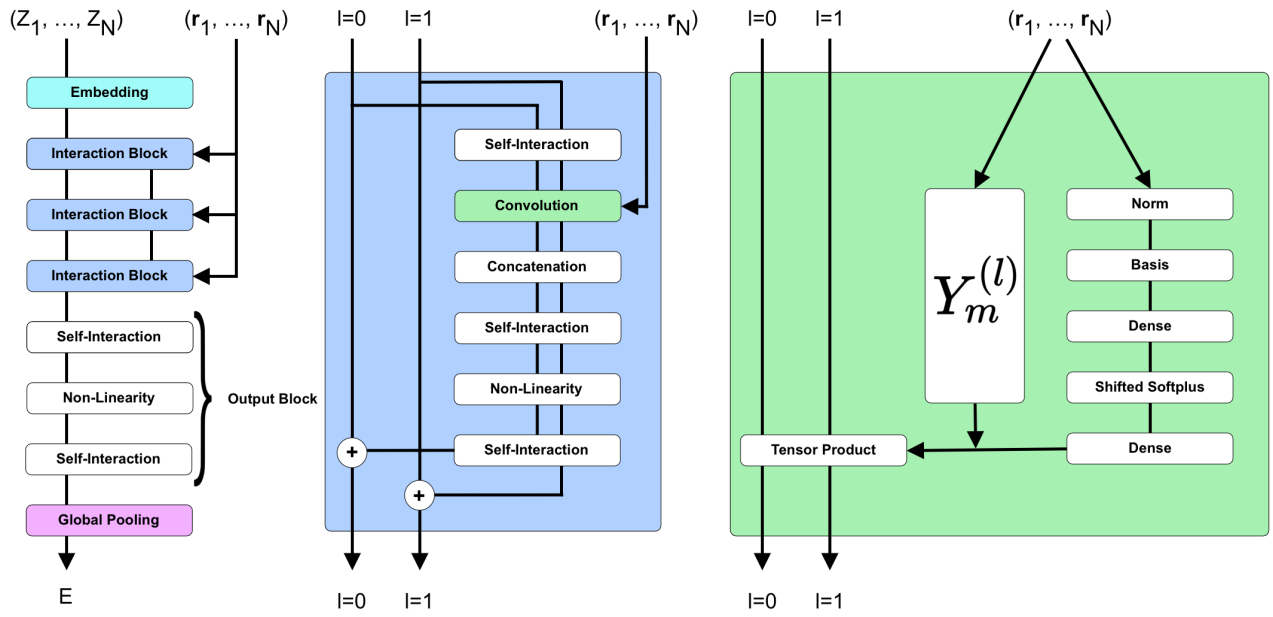

FIG. 2: The NequIP network architecture. Left: atomic numbers are embedded into $l = 0$ features, which are refined through a series of interaction blocks, creating $l = 0$ and $l = 1$ features. An output block generates atomic energies, which are pooled to give the total predicted energy. Middle: the interaction block consists of a series of convolutions, interweaved with self-interaction layers, equivariant nonlinearities and concatenation. Right: the convolution combines the radial function $R(r)$ which operates only on interatomic distances with the spherical harmonics based on unit vector $\hat{r}$ via a tensor product.

MD-17 small molecule dynamics

We first evaluate NequIP on MD-17 [17, 30, 31], a data set of eight small organic molecules in which reference values of energy and forces are generated by ab-initio MD simulations with DFT. For training we use N=1,000 structure configurations for each molecule, sampled uniformly from the full data set, the same number of configurations for validation, and evaluate the test error on all remaining configurations in the data set. The mean absolute error in the force components is shown in Table I in units of [meV/Å]. We compare results using NequIP with those from published leading ML-IP models that were also trained on 1,000 structures: in particular SchNet [9], DimeNet [11] (both graph neural networks), sGDML [31], and FCHL19/GPR (kernel-based methods) [33]. We find that NequIP outperforms SchNet and sGDML on all molecules in the data set, DimeNet on 7 out of 8 molecules (on par on the remaining one), and performs on par with FCHL/19GPR. The consistent improvement in accuracy upon sGDML and the comparable performance to FCHL19/GPR are particularly surprising, as these are based on kernel methods, that typically tend to be more sample efficient. It should be noted, however, that the evaluation cost of kernel methods scales linearly with the number of training configurations. Note also that on some molecules, NequIP trained on 1,000 configurations even performs as well as SchNet trained on 50,000 structures [9]: on aspirin and naphthalene, for example, the NequIP network trained on 1,000 structures produces mean absolute errors in the forces of 15.1 meV/Å and 4.2 meV/Å, respectively, compared to 14.3 meV/Åand 4.8 meV/Å of SchNet trained on 50x more molecules, hinting that NequIP exhibits exceptional data efficiency. On other molecules such as ethanol, however, SchNet trained with 50,000 molecules still clearly outperforms NequIP trained with 1,000 molecules (2.2 meV/Å for SchNet for N=50,000 vs 9.0 meV/Å for NequIP for N=1,000).

Force training at quantum chemical accuracy

Ability to achieve high accuracy on a comparatively small data set opens the door to training models on expensive high-order *ab-initio* quantum chemical methods. It has been shown that DFT can fail to capture important subtleties in the potential energy surface, potentially even identifying the wrong ground states [31]. This problem can be remedied through the use of more accurate reference calculations, such as coupled cluster methods CCSD(T), typically regarded

Molecule	NequIP	SchNet	sGDML	DimeNet	FCHL19/GPR
Aspirin	15.1	58.5	29.5	21.6	20.7
Benzene [17]	8.1	13.4	n/a	8.1	n/a
Benzene [31]	2.3	n/a	2.6	n/a	n/a
Ethanol	9.0	16.9	14.3	10.0	5.9
Malonaldehyde	14.6	28.6	17.8	16.6	10.6
Naphthalene	4.2	25.2	4.8	9.3	6.5
Salicylic Acid	10.3	36.9	12.1	16.2	9.6
Toluene	4.4	24.7	6.1	9.4	8.8
Uracil	7.5	24.3	10.4	13.1	4.6

TABLE I: MAE of force components on the MD-17 data set, trained on 1,000 configurations, forces in units of [meV/Å]. For the benzene molecule, two different data set exists from [17], [31] with different levels of accuracy in the DFT reference data.

as the gold standard of quantum chemistry. However, the high computational cost of CCSD(T) has thus far hindered the use of reference data structures at this level of theory, prohibited by the need for large data sets that are required by available NN-IPs. Leveraging the high data efficiency of NequIP, we evaluate it on a set of molecules computed at quantum chemical accuracy (aspirin at CCSD, all others at CCSD(T)) [31] and compare the results to those reported for sGDML [31]. The training/validation set consists of a total of 1,000 molecular structures which we split into 950 for training and 50 for validation (sampled uniformly), and we test the accuracy on all remaining structures (we use the train/test split provided with the data set, but further split the training set into training and validation sets). We find that NequIP achieves lower errors on four out of five molecules, performing on par with sGDML on the fifth molecule, as shown in Table II.

Molecule	NequIP	sGDML
Aspirin	14.7	33.0
Benzene	0.8	1.7
Ethanol	9.4	15.2
Malonaldehyde	16.0	16.0
Toluene	4.4	9.1

TABLE II: Force MAE for molecules at CCSD/CCSD(T) accuracy, reported in units of [meV/Å], with 1,000 reference configurations).

Liquid Water and Ice Dynamics

To demonstrate the applicability of NequIP beyond small molecules, we evaluate the method on a series of extended systems with periodic boundary conditions. As a first example we use a joint data set consisting of liquid water and three ice structures [15, 32], computed at the PBE0-TS level of theory. This data set contains [15]: a) liquid water, P=1bar, T=300K, computed via path-integral AIMD, b) ice Ih, P=1bar, T=273K, computed via path-integral AIMD c) ice Ih, P=1bar, T=330K, computed via classical AIMD d) ice Ih, P=2.13 kbar, T=238K, computed via classical AIMD. The liquid water system consists of 64 H_2O molecules (192 atoms), while the ice structures consist of 96 H_2O molecules (288 atoms). A DeepMD NN-IP model was previously trained [15] for water and ice using a joint training set containing 133,500 reference calculations of these four systems. To assess data efficiency of the NequIP architecture, we similarly train a model jointly on all four parts of the data set, but using only 133 structures for training, i.e. 1000x fewer data. The 133 structures were sampled uniformly from the full data set available online, consisting of water and ice structures, made up of a total of 140,000 frames, coming from the same MD trajectories that were used in the earlier work [15]. We also use a validation set of 50 frames and report the test accuracy on all remaining structures in the data set. Table III shows the comparison of the predictive force accuracy of NequIP trained on the 133 structures vs DeepMD trained on 133,500 structures. We find that with 1000x fewer training data, NequIP significantly outperforms DeepMD on all four parts of the data set.

Heterogeneous catalysis of formate dehydrogenation

Next, we demonstrate application of NequIP to a catalytic surface reaction. In particular, we investigate the dynamics of formate undergoing dehydrogenation decomposition ($HCOO^* \rightarrow H^* + CO_2$) on a Cu $< 110 >$ surface (see Figure 3). This system is highly heterogeneous, with both metallic and covalent types of bonding as well as charge transfer occurring between the metal and the molecule, making this a particularly challenging test system. Different states of the molecule also lead to dissimilar C-O bond lengths [34, 35]. Training structures consist of 48 Cu atoms and 4 atoms of the molecule (HCOO* or CO_2+H*). The MAE of the predicted forces using a NequIP model trained on

System	NequIP, 133 data points	DeepMD, 133,500 data points
Liquid Water	35.9	40.4
Ice Ih (b)	25.9	43.3
Ice Ih (c)	16.6	26.8
Ice Ih (d)	13.5	25.4

TABLE III: Root mean square error (RMSE) of force components on liquid water and the three ices in units of [meV/A]. Note that the NequIP model was trained on $< 0.1\%$ of the training data of DeepMD.

Element	MAE
C	55.8
O	86.7
H	42.0
Cu	54.5
Total structure	55.6

TABLE IV: MAE of force components for Formate on Cu system, per-element basis. The training set consists of 2,500 structures, force units are [meV/A]

2,500 structures is shown in Table IV, demonstrating that NequIP is able to accurately model the interatomic forces for this complex reactive system. A more detailed analysis of the resulting dynamics will be subject of a separate study.

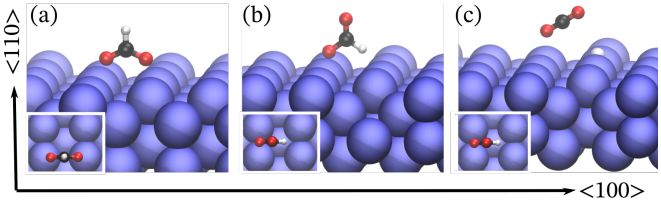

FIG. 3: Perspective view of atomic configurations of (a) bidentate HCOO (b) monodentate HCOO and (c) CO_2 and a hydrogen adatom on a Cu(110) surface. The blue, red, black, and white spheres represent Cu, O, C, and H atoms, respectively. The subset shown in each subplot is the corresponding top view along the $< 110 >$ orientation.

Lithium Phosphate Amorphous Glass Formation

To examine the ability of the model to capture dynamical properties, we demonstrate that NequIP can describe structural dynamics in amorphous lithium phosphate with composition $Li_4P_2O_7$. This material is a member of the promising family of solid electrolytes for Li-metal batteries [12, 36, 37], with non-trivial Li-ion transport and phase transformation behaviors. The training data set consists of two 50ps-long AIMD simulations, one of the molten structure at T=3000 K, followed by another of a quenched glass structure at T=600 K. We train NequIP on a subset of 1,000 structures of the molten trajectory, each consisting of 208 atoms, and sampled uniformly from the full data set of 25,000 AIMD frames. We use a validation set of 100 structures, and evaluate the model on all remaining structures. Table V shows the test set error in the force components on both the test set from the AIMD molten trajectory and the full AIMD quenched glass trajectory. To then evaluate the physical fidelity of the trained model, we use it to run a 50 ps MD simulation at T=600 K and compare the total radial distribution function (RDF) without element distinction as well as the angular distribution functions (ADF) of the P-O-O (P central atom) and O-P-P (O central atom) angles to the *ab-inito* trajectory at the same temperature. The P-O-O angle corresponds to the tetrahedral bond angle, while the O-P-P corresponds to a bridging angle between corner-sharing phosphate tetrahedra (Figure 4). Figure 5 shows that NequIP can accurately reproduce the RDF and the two ADFs, in comparison with AIMD, after training on only 1,000 structures. This demonstrates that the model generates the glass state and recovers

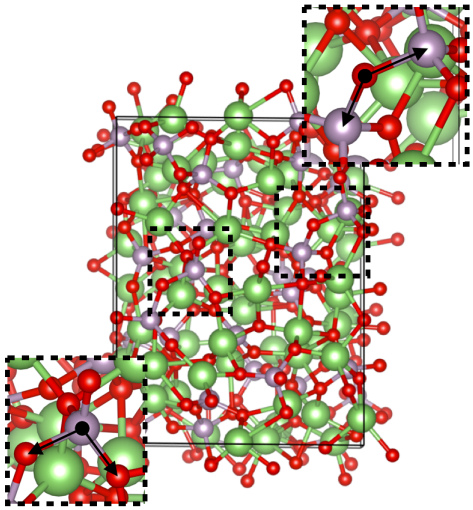

FIG. 4: Quenched glass structure of $Li_4P_2O_7$. The insets show the P-O-O tetrahedral bond angle (bottom left) as well as the O-P-P bridging angle between corner-sharing phosphate tetrahedra (top right).

its dynamics and structure almost perfectly, having seen only the high-temperature molten training data.

Lithium Thiophosphate Superionic Transport

To show that NequIP can model kinetic transport properties from small training sets at high accuracy, we study Li-ion diffusivity in LiPS ($Li_{6.75}P_3S_{11}$) a crystalline superionic Li conductor, consisting of a simulation cell of 83 atoms [13]. MD is widely used to study diffusion; however, training a ML-IP to the accuracy required to accurately predict kinetic properties has in the past required large training set sizes ([38] e.g. uses a data set of 30,874 structures to study Li diffusion in Li_3PO_4). Here we demonstrate that not only does NequIP obtain small errors in the force components, but it also accurately predicts the diffusivity after training on a data set obtained from an AIMD simulation. Again, we find that very small training sets lead to highly accurate models, as shown in Table V for training set sizes of 10, 100, 1,000 and 2,500 structures. We run a series of MD simulations with the NequIP potential trained on 2,500 structures in the NVT ensemble at the same temperature as the AIMD simulation for a total simulation time of 50 ps and a time step of 0.25 fs, which we found advantageous for reliability and stability of long simulations. We measure the Li diffusivity in ten Nequip-driven MD simulations (computed via the slope of the mean square displacement), all of length 50 ps and started from different initial velocities, randomly sampled from a Maxwell-Boltzmann distribution. We find a mean diffusivity of $1.42 \times 10^{-5}\,\mathrm{cm}^2/\mathrm{s}$, in excellent agreement with the diffusivity of $1.38 \times 10^{-5}\,\mathrm{cm}^2/\mathrm{s}$ computed from AIMD, thus achieving a relative error of as little as 3%. Figure 6 shows the mean square displacements of Li for an example run.

System	Data Set Size	MAE
LiPS	10	157.1
LiPS	100	50.0
LiPS	1,000	25.1
LiPS	2,500	24.1
$Li_4P_2O_7$, melt	1,000	63.2
$Li_4P_2O_7$, quench	1,000	36.9

TABLE V: Force MAE for LiPS and $Li_4P_2O_7$ for different data set sizes in units of [meV/A]. The model for $Li_4P_2O_7$ was trained exclusively on structures from the melted trajectory, the reported test errors show the MAE on both the test set of the melted trajectory as well as the full quench trajectory.

Data Efficiency

In the above experiments, NequIP exhibits exceptionally high data efficiency, i.e. it can be trained successfully to state-of-the-art accuracy from unexpectedly small training sets. It is interesting to consider the reasons for such high performance and verify that it is connected to the equivariant nature of the model. First, it is important to note that each training configuration contains multiple labels, thus increasing the total number of labels available beyond just the potential energy label associated with each structure. In particular, for a training set of M first-principles calculations with structures consisting of N atoms, the total number of labels available is $M(3N+1)$ since every force component on every atom constitutes a label and so does the total energy of the reference calculation (we only train to atomic forces and not energies, thus using $3MN$ force components as labels).

In order to gain insight into the reasons behind increased accuracy and data efficiency, we perform a series of experiments with the goal of isolating the effect of using equivariant convolutions of geometric tensors compared to invariant convolutions over scalars. In particular, we run a set of experiments for a system with a fixed number of training configurations in which we explicitly turn on or off interactions of higher order than $l = 0$. This defines two settings: first, we train the network with both $l = 0$ and $l = 1$ features and all five interactions as previously outlined in this work. Second, when all interactions involving $l = 1$ are turned off, this turns the network into a conventional invariant GNN-IP, involving only invariant convolutions over scalar features in a SchNet-style fashion. As a test system we chose bulk water: in particular we use the data set introduced in [39], consisting of 1,593 bulk liquid water structures with 64 water molecules each. We train a series of networks with identical hyperparameters, but vary the training set sizes between 10 and 1,000 structures, sampled uniformly from the full data set, as well as a validation set consisting of 100 structures. We then evaluate the error on all remaining structures for a given training set size. As shown in Figure 7, we find that the equivariant setting (using $l = 0$ and $l = 1$) significantly outperforms the invariant setting (using only $l = 0$) for all data set sizes as measured by the MAE of force components. This suggests that it is indeed the use of tensor (in our specific case vector) features and equivariant convolutions that enables the high data efficiency of NequIP. We further note, that in [39], a Behler-Parrinello Neural Network (BPNN) was trained on 1303 structures, yielding a RMSE of $\approx$ 120 meV/Å in forces when evaluated on the remaining 290 structures. We find that NequIP models trained with as little as 50 and 100 data points obtain RMSEs of 122.9 meV/Å and 93.3 meV/Å on their

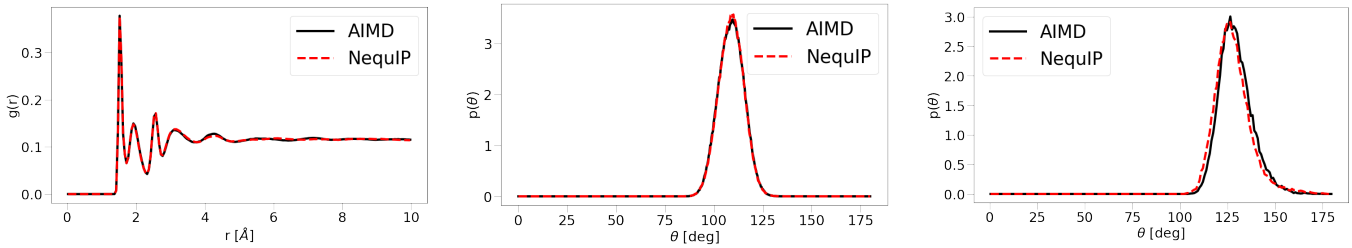

FIG. 5: Left: Radial Distribution Function, middle: Angular Distribution Function, bridging oxygen, right: Angular Distribution Function, tetrahedral bond angle. All are defined as probability density functions.

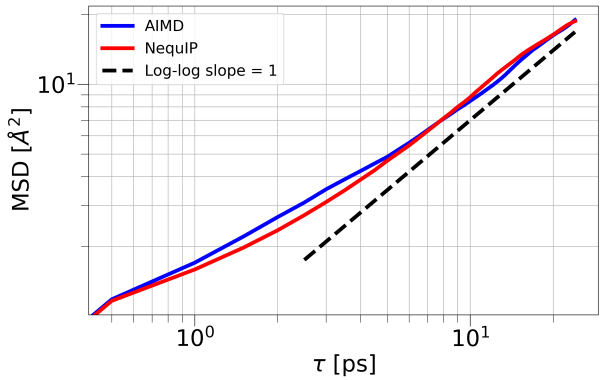

FIG. 6: Comparison of Lithium mean square displacement of AIMD and NequIP trajectories.

respective test sets (note that Figure 7 shows the MAE). This provides further evidence that NequIP exhibits significantly improved data efficiency in comparison with existing methods.

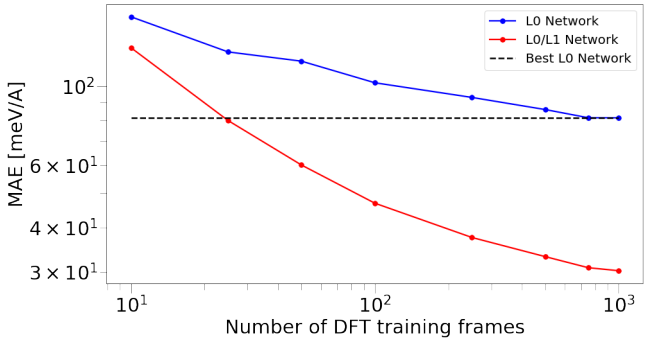

FIG. 7: Log-log plot of the predictive error in forces of NequIP with $l = 0$ vs. $l = 0/l = 1$ interactions as a function of data set size, measured via the force MAE.

Computational Efficiency

Finally, we report the computational efficiency of NequIP and compare it to that of the *ab-inito* methods on two examples shown in this work: for a molecular system, we choose the Toluene molecule, computed at the CCSD(T)-level of theory [31]; for a material with periodic boundary conditions, we choose the Formate on Cu system, in which reference data were obtained with DFT. For both systems, we report the time required for a single force call on a CPU node with 32 cores. The results are shown in Table VI. In both cases, NequIP gives a large speed-up over the *ab-initio* methods. In the case of the Toluene system, this means that 58.4 minutes of a NequIP simulation can obtain the simulation time equaling one century of a CCSD(T) simulation.

DISCUSSION

We demonstrate that the Neural Equivariant Interatomic Potential (NequIP), a new type of graph neural network built on SE(3)-equivariant convolutions exhibits state-of-the-art accuracy and exceptional data efficiency on data sets of small molecules and periodic materials. Furthermore, we find that we can reproduce structural and kinetic properties from molecular dynamics simulations with very high fidelity in comparison to ab-initio simulations. The ability to both learn from small numbers of reference samples, while retaining high computational efficiency opens the door to performing simulations of large systems over long time-scales at quantum mechanical accuracy, using DFT or higher order methods such as coupled-cluster or quantum Monte Carlo data as reference. We expect the new method will enable researchers in computational chemistry, physics, biology, and materials science to conduct molecular dynamics simulations of complex reactions and phase transformations at increased accuracy and efficiency.

System	Number of atoms	NequIP	Ab-initio	Speed-up
Toluene	15	16 ms	4 hours*	900,000
Formate on Cu	52	58 ms	1045.6 s	18,028

TABLE VI: Time required for a single force call for NequIP in comparison to CCSD(T) for Toluene and DFT for Formate on Cu; * personal communication with Stefan Chmiela and Alexandre Tkatchenko.

METHODS

Reference Data Sets

MD-17: MD-17 [17, 30, 31] is a data set of eight small organic molecules, obtained from MD simulations at T=500K and computed at the PBE+vdW-TS level of electronic structure theory, resulting in data set sizes between 133,770 and 993,237 structures. The data set was obtained from `http://quantum-machine.org/gdml/#datasets`.

Molecules@CCSD/CCSD(T): The data set of small molecules at CCSD and CCSD(T) accuracy [31] contains positions, energies, and forces for five different small molecules: Asprin (CCSD), Benzene, Malondaldehyde, Toluene, Ethanol (all CCSD(T)). Each data set consists of 1,500 structures with the exception of Ethanol, for which 2,000 structure are available. For more detailed information, we direct the reader to [31]. The data set was obtained from `http://quantum-machine.org/gdml/#datasets`.

Liquid Water and Ice: The data set of liquid waters and ice structures [15, 32] was generated from classical AIMD and path-integral AIMD simulations at different temperatures and pressures, computed with a PBE0-TS functional [15]. The data set, obtained from `http://www.deepmd.org/database/deeppot-se-data/`, contains a total of 140,000 structures, of which 100,000 are liquid water and 20,000 are Ice Ih b),10,000 are Ice Ih c), and another 10,000 are Ice Ih d).

Formate decomposition on Cu: The decomposition process of formate on Cu involves configurations corresponding to the cleavage of the C-H bond, initial and intermediate states (monodentate, bidentate formate on Cu $<$ 110 $>$) and final states (H ad-atom with a desorbed CO_2 in the gas phase). Nudged elastic band (NEB) method was first used to generate an initial reaction path of the C-H bond breaking. 12 short ab initio molecular dynamics, starting from different NEB images, were run to collect a total of 6855 DFT structures. The CP2K [40] code was employed for the AIMD simulations. Each trajectory was generated with a time step of 0.5 fs and 500 total steps. We train NequIP on 2,500 reference structures sampled uniformly from the full data set of 6,855 structures, use a validation set of 250 structures and evaluate the mean absolute error on all remaining structures. Due to the unbalanced nature of the data set (more atoms of Cu than in the molecule), we use a per-element weighed loss function in which atoms C, O_1, O_2, and H and the sum of all Cu atoms all receive equal weights.

$Li_4P_2O_7$ glass: The $Li_4P_2O_7$ ab-initio data were generated using an ab-initio melt-quench MD simulation, starting with a stoichiometric crystal of 208 atoms (space group P21/c) in a periodic box of $10.4 \times 14.0 \times 16.0$ Å. The dynamics used the Vienna Ab-Initio Simulation Package (VASP) [41–43], with a generalized gradient PBE functional [44], projector augmented wave (PAW) pseudopotentials [45], a Nosé-Hoover thermostat, a time step of 2 fs, a plane-wave cutoff of 400 eV, and a Γ-point reciprocal-space mesh. The crystal was melted at 3000 K for 50 ps, then immediately quenched to 600 K and run for another 50 ps. The resulting structure was confirmed to be amorphous by plotting the radial distribution function of P-P distances. The training was performed only on the molten portion, and the MD simulations for a quenched simulation.

LiPS: Lithium phosphorus sulfide (LiPS) based materials are known to exhibit high lithium ion conductivity, making them attractive as solid-state electrolytes for lithium-ion batteries. Other examples of known materials in this family of superionic conductors are LiGePS and LiCuPS-based compounds. The training data set is taken from a previous study on graph neural network force field [13], where the LiPS training data were generated using ab-initio MD of an LiPS structure with Li-vacancy ($Li_{6.75}P_3S_{11}$) consisting of 27 Li, 12 P, and 44 S atoms respectively. The structure was first equilibrated and then run at 520 K using the NVT ensemble for 50 ps with a 2.0 fs time step. The full data set contains 25,001 MD frames. We set aside 10,000 frames as a fixed test set. From the remaining frames, we choose training set sizes of 10, 100, 1,000, and 2,500 frames with a fixed validation set size of 100. In order to generate a diverse training set, we sample both the training and validation sets in a way such that 30% of both of them are comprised of the structures with the shortest interatomic distances out of all frames not in the test set and the remaining 70% of the training and validation set are uniformly sampled.

Liquid Water, Cheng et al.: The training set used in the data efficiency experiments on water consists of 1,593 reference calculations of bulk liquid water at the revPBE0-D3 level of accuracy, with each structure containing 192 atoms, as given in [39]. Further information can be found in [39]. The data set was obtained from `https://github.com/BingqingCheng/ab-initio-thermodynamics-of-water`.

Molecular Dynamics Simulations. To run MD simulations, NequIP force outputs were integrated with the Atomic Simulation Environment (ASE) [27] in which we implement a custom version of the Nosé-Hoover thermostat. We use this in-house implementation for the both the $Li_4P_2O_7$ as well as the LiPS MD simulations. The thermostat parameter was chosen to match the temperature fluctuations observed in the AIMD run.

Training. Networks are trained using a loss function based on atomic forces:

$$\mathcal{L} = \frac{1}{3N}\sum_{i=1}^{N}\sum_{\alpha=1}^{3} || - \frac{\partial \hat{E}}{\partial r_{i,\alpha}} - F_{i,\alpha}||^2 \qquad (11)$$

where N is the number of atoms in the system and $\hat{E}$ is the predicted potential energy. Note that we do not train on energies since atomic forces are the only quantities required to integrate Newton's equations of motion. Since the predicted forces are computed as the gradient of a scalar potential, they are still conservative. If energies are of interest, however, one can add them to the loss function and determine the relative weighting via a trade-off parameter as done in previous works [9, 11]. In a similar fashion, it is trivial to add other quantities of interest to the loss function (e.g predicting atomic charges or multipole tensors can be of interest for modeling long-range interactions), where they may be scalar fields, vector fields, or higher-order tensor fields.

Hyperparameters. Training of models was performed on NVIDIA Tesla V100 GPUs. Throughout all experiments shown in this work, we use a feature dimension of $h = 64$, 6 interaction blocks, $N_b = 8$ Bessel basis functions and radial neural networks with one hidden layer, also of hidden dimension $N_{hidden} = 8$, giving light-weight radial functions with a comparatively small number of parameters. The final interaction block is followed by the output block, which first reduces the feature dimension to 16 through a self-interaction layer. An equivariant non-linearity is applied and finally through another self-interaction layer the feature dimension is reduced to a single scalar output value associated with each atom that is then summed over to give the total potential energy. In all experiments, we use the Adam optimizer [46] with the TensorFlow 1.14 default settings of $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We decrease the initial learning rate of 0.001 by a decay factor of 0.8 whenever the validation RMSE in the forces has not seen an improvement for a given number of epochs: for the small molecule tasks, we set this learning rate patience to 1,000, for all other tasks we use 100. We continuously save the model with the best validation RMSE and use the model with the overall best RMSE for evaluation on the test set and MD simulations. We stop the training if either a maximum number of 50,000 epochs (one epochs equals a full pass over the training set) has been reached, or the validation force RMSE has not improved for 2,500 epochs, or the maximum training time has been exceeded, whichever occurs first. All systems were trained for a maximum of 8 days (consisting of four runs of 48-hour time-limited compute jobs, which are restarted from the best saved model, i.e. potentially including repeats in the training) with the exception of the $Li_4P_2O_7$, which was trained for 12 days (six 48-hour compute jobs) and the LiPS systems, which were trained for 4 days (two 48-hour compute jobs). We use a batch size of 5 structures for all small molecule tasks, and a batch size of 1 structure for all other tasks. We found small batch sizes to be important for obtaining high predictive accuracy. We also found it important to choose the radial cutoff distance r_c appropriately. A list of the cutoff radii in units of [Å] that were used for the different systems is given in Table VII.

Data Set	Cutoff
MD-17 [17, 30, 31]	4.0
Molecules, CCSD/CCSD(T) [31]	4.0
Water+Ices, DeepMD [15, 32]	6.0
Formate on Cu	5.0
$Li_4P_2O_7$ [12]	5.0
LiPS [13]	5.0
Water, data efficiency tests [39]	4.5

TABLE VII: Radial cutoff in units of [Å].

DATA AVAILABILITY

The code and data sets will be made available upon publication.

AUTHOR CONTRIBUTIONS

S.B. initiated the project, conceived the NequIP model, implemented the software and conducted all software experiments under the guidance of B.K. T.E.S. contributed to the conception of the model, guidance of computational experiments, and the

software implementation. L.S. created the data set for formate/Cu, guided work on training and MD simulations on this system, and contributed to development of the software implementation. J.P.M. guided the work on the LiPS conductor and implemented the thermostat for MD simulations together with S.B.. M.K. created the AIMD data set of $Li_4P_2O_7$, wrote software for the analysis of MD results and guided the benchmarking on this system. N.M. wrote software for the computation of the diffusion results and guided discussions on the interpretation of results. B.K. supervised the project from conception to design of experiments, implementation, theory, as well as analysis of data. All authors contributed to the manuscript and the discussion of results.

ACKNOWLEDGEMENTS

We thank Jonathan Vandermause, Cheol Woo Park, David Clark, Kostiantyn Lapchevskyi, Mario Geiger, Joshua Rackers, and Benjamin Kurt Miller for helpful discussions.

We thank Stefan Chmiela and Alexandre Tkatchenko for providing the timing data for the Toluene system.

Work at Harvard was supported by Bosch Research and the Integrated Mesoscale Architectures for Sustainable Catalysis (IMASC), an Energy Frontier Research Center funded by the US Department of Energy (DOE), Office of Science, Office of Basic Energy Sciences under Award No. DE-SC0012573. N.M. acknowledges support from the Department of the Navy, Office of Naval Research.

Work at Bosch Research was partially supported by ARPA-E Award No. DE-AR0000775 and used resources of the Oak Ridge Leadership Computing Facility at Oak Ridge National Laboratory, which is supported by the Office of Science of the Department of Energy under Contract DE-AC05-00OR22725.

T.E.S. was supported by the Laboratory Directed Research and Development Program of Lawrence Berkeley National Laboratory under U.S. Department of Energy Contract No. DE-AC02-05CH11231.

The authors acknowledge computing resources provided by the Harvard University FAS Division of Science Research Computing Group and by the Texas Advanced Computing Center (TACC) at The University of Texas at Austin under allocations DMR20009 and DMR20013.

[1] W. D. Richards, T. Tsujimura, L. J. Miara, Y. Wang, J. C. Kim, S. P. Ong, I. Uechi, N. Suzuki, and G. Ceder, Nature communications **7**, 1 (2016).

[2] M. Boero, M. Parrinello, and K. Terakura, Journal of the American Chemical Society **120**, 2746 (1998).

[3] K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw, Science **334**, 517 (2011).

[4] J. Behler and M. Parrinello, Physical review letters **98**, 146401 (2007).

[5] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, Physical review letters **104**, 136403 (2010).

[6] A. V. Shapeev, Multiscale Modeling & Simulation **14**, 1153 (2016).

[7] A. P. Thompson, L. P. Swiler, C. R. Trott, S. M. Foiles, and G. J. Tucker, Journal of Computational Physics **285**, 316 (2015).

[8] J. Vandermause, S. B. Torrisi, S. Batzner, Y. Xie, L. Sun, A. M. Kolpak, and B. Kozinsky, npj Computational Materials **6**, 1 (2020).

[9] K. Schütt, P.-J. Kindermans, H. E. S. Felix, S. Chmiela, A. Tkatchenko, and K.-R. Müller, in *Advances in neural information processing systems* (2017) pp. 991–1001.

[10] O. T. Unke and M. Meuwly, Journal of chemical theory and computation **15**, 3678 (2019).

[11] J. Klicpera, J. Groß, and S. Günnemann, arXiv preprint arXiv:2003.03123 (2020).

[12] J. P. Mailoa, M. Kornbluth, S. Batzner, G. Samsonidze, S. T. Lam, J. Vandermause, C. Ablitt, N. Molinari, and B. Kozinsky, Nature machine intelligence **1**, 471 (2019).

[13] C. W. Park, M. Kornbluth, J. Vandermause, C. Wolverton, B. Kozinsky, and J. P. Mailoa, arXiv preprint arXiv:2007.14444 (2020).

[14] N. Artrith and A. M. Kolpak, Nano letters **14**, 2670 (2014).

[15] L. Zhang, J. Han, H. Wang, R. Car, and E. Weinan, Physical review letters **120**, 143001 (2018).

[16] J. S. Smith, O. Isayev, and A. E. Roitberg, Chemical science **8**, 3192 (2017).

[17] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, Science advances **3**, e1603015 (2017).

[18] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, arXiv preprint arXiv:1704.01212 (2017).

[19] B. Anderson, T. S. Hy, and R. Kondor, in *Advances in Neural Information Processing Systems* (2019) pp. 14537–14546.

[20] R. J. Townshend, B. Townshend, S. Eismann, and R. O. Dror, arXiv preprint arXiv:2006.14163 (2020).

[21] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley, arXiv preprint arXiv:1802.08219 (2018).

[22] A. Grisafi, D. M. Wilkins, M. J. Willatt, and M. Ceriotti, in *Machine Learning in Chemistry: Data-Driven Algorithms, Learning Systems, and Predictions* (ACS Publications, 2019) pp. 1–21.

[23] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, arXiv preprint arXiv:1603.04467 (2016).

[24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, in *Advances in neural information processing systems* (2019) pp. 8026–8037.

[25] M. Weiler, M. Geiger, M. Welling, W. Boomsma, and T. S. Cohen, in *Advances in Neural Information Processing Systems* (2018) pp. 10381–10392.

[26] R. Kondor, Z. Lin, and S. Trivedi, in *Advances in Neural Information Processing Systems* (2018) pp. 10117–10126.

[27] A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. SchÃŒtt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, Journal of Physics: Condensed Matter **29**, 273002 (2017).

[28] T. E. Smidt, M. Geiger, and B. K. Miller, arXiv preprint arXiv:2007.02005 (2020).

[29] K. He, X. Zhang, S. Ren, and J. Sun, in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016) pp. 770–778.

[30] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, Nature communications **8**, 1 (2017).

[31] S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko, Nature communications **9**, 1 (2018).

[32] H.-Y. Ko, L. Zhang, B. Santra, H. Wang, W. E, R. A. DiStasio Jr, and R. Car, Molecular Physics **117**, 3269 (2019).

[33] A. S. Christensen, L. A. Bratholm, F. A. Faber, and O. Anatole von Lilienfeld, The Journal of Chemical Physics **152**, 044107 (2020).

[34] W. S. Sim, P. Gardner, and D. A. King, The Journal of Physical Chemistry **100**, 12509–12516 (1996), 00000.

[35] G. Wang, Y. Morikawa, T. Matsumoto, and J. Nakamura, The Journal of Physical Chemistry B **110**, 9–11 (2006), 00050.

[36] X. Yu, J. B. Bates, G. E. Jellison, and F. X. Hart, Journal of The Electrochemical Society **144**, 524 (1997).

[37] A. S. Westover, A. K. Kercher, M. Kornbluth, M. Naguib, M. J. Palmer, D. A. Cullen, and N. J. Dudney, ACS Applied Materials & Interfaces **12**, 11570 (2020).

[38] W. Li, Y. Ando, E. Minamitani, and S. Watanabe, The Journal of chemical physics **147**, 214106 (2017).

[39] B. Cheng, E. A. Engel, J. Behler, C. Dellago, and M. Ceriotti, Proceedings of the National Academy of Sciences **116**, 1110 (2019).

[40] J. Hutter, M. Iannuzzi, F. Schiffmann, and J. VandeVondele, WIREs Computational Molecular Science **4**, 15–25 (2014), 00000.

[41] G. Kresse and J. Hafner, Physical Review B **47**, 558 (1993).

[42] G. Kresse and J. Furthmüller, Computational Materials Science **6**, 15 (1996).

[43] G. Kresse and J. Furthmüller, Physical Review B **54**, 11169 (1996).

[44] J. P. Perdew, K. Burke, and M. Ernzerhof, Physical Review Letters **77**, 3865 (1996).

[45] G. Kresse and D. Joubert, Physical Review B **59**, 1758 (1999).

[46] D. P. Kingma and J. Ba, arXiv preprint arXiv:1412.6980 (2014).