NewtonNet: A Newtonian message passing network for deep learning of interatomic potentials and forces

Mojtaba Haghighatlari[1], Jie Li[1], Xingyi Guan[1,5], Oufan Zhang[1],
Akshaya Das[1],Christopher J. Stein[1,5−6],Farnaz Heidar-Zadeh[1−3],Meili Liu[1,4],
Martin Head-Gordon[1,5],Luke Bertels[1], Hongxia Hao[1,5],Itai Leven[1,5],
Teresa Head-Gordon[1,5,7]

[1]Kenneth S. Pitzer Theory Center and Department of Chemistry, University of California, Berkeley, CA, USA
[2]Center for Molecular Modeling (CMM), Ghent University, B-9052 Ghent, Belgium
[3]Department of Chemistry, Queen's University, Kingston, Ontario K7L 3N6, Canada
[4]Department of Chemistry, Beijing Normal University, Beijing 100875, China
[5]Chemical Sciences Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA
[6] Theoretical Physics and Center for Nanointegration Duisburg-Essen (CENIDE), University of Duisburg-Essen, 47048 Duisburg, Germany
[7]Departments of Bioengineering and Chemical and Biomolecular Engineering, University of California, Berkeley, CA, USA

corresponding author: thg@berkeley.edu

Abstract

We report a new deep learning message passing network that takes inspiration from Newton's equations of motion to learn interatomic potentials and forces. With the advantage of directional information from trainable force vectors, and physics-infused operators that are inspired by Newtonian physics, the entire model remains rotationally equivariant, and many-body interactions are inferred by more interpretable physical features. We test NewtonNet on the prediction of several reactive and non-reactive high quality *ab initio* data sets including single small molecules, a large set of chemically diverse molecules, and methane and hydrogen combustion reactions, achieving state-of-the-art test performance on energies and forces with far greater data and computational efficiency than other deep learning models.

INTRODUCTION

The combinatorial size of chemical reaction space, which compounds under variable synthetic, catalytic, and/or non-equilibrium conditions, is vast. This makes application of first principles quantum mechanical and advanced statistical mechanics sampling methods to identify all reaction pathways challenging, even when considering better physics-based models, algorithms, or future exascale computing paradigms. If we could develop new machine learning approaches to chemical reactivity, we would be able to better tackle many fascinating but quite difficult chemical systems ranging from metal organic frameworks for binding CO_2 from air or H_2 for hydrogen storage, mechanistic studies of enzymes that accelerate biological reactions, the reactive chemistry at the solid-liquid interface in electrocatalysis, and developing new catalysts that are highly selective and which exhibit stereo-, regio-, and chemo-selectivity[1–4].

The modernization of machine learning as applied to the chemical sciences can be traced to the artificial neural network (ANN) representation by Behler and Parrinello[5] to describe the high dimensional potential energy surfaces (PES) important to chemical reactivity. Their first realization is that the intrinsic description of energies or forces that depend on Cartesian variables needs to be replaced by the use of localized Gaussian symmetry functions that invoke permutation, rotational, and translational invariance to data representations for learning potential energy surfaces. To be more specific, the energy of an atomic configuration should be invariant to a global rotation when presented to the ANN. In addition, these symmetry functions are made many-bodied through their stacking, with data presentation utilizing 50 symmetry functions with different learnable parameters of each atom's chemical environment.

Alternatively message passing neural networks[6] (MPNN) have emerged that replaces the hand-crafted features of the distances and angles of symmetry functions with trainable operators that only rely on the atomic Z-numbers and positions to learn the representations of the heterogeneous chemical environment directly from the training data.[7] A major contributing MPNN method for 3D structures is SchNet[7], which takes advantage of the convolution of decomposed interatomic distances with atomic attributes, and related methods have subsequently built on this success in incorporating additional features to describe atomic environments. For example, PhysNet[8] adds prior knowledge about the long-range electrostatics in energy predictions, and DimeNet[9] takes advantage of angular information and more stable basis functions based on Bessel functions.

But in the standard MPNN, the representation is usually reduced to transformationally identical features, for example quantities that are invariant to translation and permutation such as the energy. However, we aim to predict not only energies but force vectors, and given the fact that vectorial features can be affected by transformation of the input structure, we need to ensure the output of each operator also will reflect such transformation equivalently when needed. More specifically, rotational transformations (such as through angular displacements) are one of the biggest challenges in the modeling of 3D objects, illustrated in learning a global orientation of structures for MD trajectories with many molecules, that is very difficult or infeasible.

Only very recently have machine learning methods been developed that are equivariant to the transformations in Euclidean space, and are emerging as state-of-the-art ML methods in predictive performance when evaluated on a variety of tasks that are fast superseding invariant-only models. Furthermore, equivariant models are found to greatly reduce the need for excessively large quantities of reference data, ushering in a new era for machine learning on the highest quality but also the most expensive of *ab initio* data. For instance, a group of machine learning models have introduced multipole expansions such as used in NequIP[10–13], or are designed to take advantage of precomputed features and/or higher-order tensors using molecular orbitals[14,15], while PaiNN[16] is a MPNN model that satisfies equivariance. In spite of the added advantage of infusing extra physical knowledge into machine learning models, the computational cost of spherical harmonics and availability/versatility of pre-computed features, or lack of physical interpretability, can be limiting.

In this work we introduce a geometric MPNN[17] based on Newton's equations of motion that achieves equivariance with respect to physically relevant rotational permutations. NewtonNet improves the capacity of structural information in the ML model by creating latent force vectors based on the Newton's third law. The force direction helps to describe the influence of neighboring atoms on the central atom based on their directional positions of atoms in the 3D space with respect to each other. Since we now introduce vector features as one of the attributes of atoms, we thereby enforce the model to remain equivariant to the rotations in the 3D coordinate space and preserve this feature throughout the network. By infusing more physical priors into the network architecture, NewtonNet realizes a computational cost that is more favorable, and enabling modeling of reactive and non-reactive chemistry with superior performance to currently popular machine learning methods used in chemical applications, and doing so with reductions down to only 1-10% of the original training data set sizes needed for invariant-only ML models. The importance of a large reduction in data requirements means that ML predictions of gold standard chemical theory such as hybrid DFT functionals or CCSD(T) in the complete basis set limit[18] are now more accessible for accurate PES generation needed for chemical reactivity using deep learning approaches.

NewtonNet Method

Given a molecular graph $\mathcal{G}$ with atomic features $a_i \in \mathbb{R}^{\mathrm{nf}}$ (where nf is the number of features) and interatomic attributes $e_{ij} \in \mathbb{R}^{\mathrm{b}}$, a message passing layer can be defined as[6]:

$$m_{ij} = M_l\left(a_i^l, a_j^l, e_{ij}\right) \tag{1}$$

$$m_i = \sum_{j \in \mathcal{N}(i)} m_{ij} \tag{2}$$

$$a_i^{t+1} = U_l\left(a_i^l, m_i\right) \tag{3}$$

where M_l is the message function and U_l is called the update function, and the sub-/super-script l accounts for the number of times the layer operates iteratively. A combination of explicit differentiable functions and operators with trainable parameters are the common choice for M_l and U_l. The core idea behind the iterative message passing of the atomic environments is to update the feature array a_i^t that represent each atom in its immediate environment.

NewtonNet considers a molecular graph defined by atomic numbers $Z_i \in \mathbb{R}^1$ and relative position vectors $\overrightarrow{r_{ij}} = \overrightarrow{r_j} - \overrightarrow{r_i} \in \mathbb{R}^3$, as input and applying operations that are inspired by Newton's equations of motion to create features arrays $a_i \in \mathbb{R}^{\mathrm{nf}}$ that represent each atom in its immediate environment with edges defined by force and displacement vectors, $\boldsymbol{f}$ and $\boldsymbol{dr}$, respectively, (Fig. 1a). NewtonNet takes advantage of multiple layers of message passing which are rotationally equivariant, described in detail below, in which each layer consists of multiple modules that include operators to construct force and displacement feature vectors, which are contracted to the feature arrays via the energy calculator module (Fig. 1b). We emphasize the critical role of projecting equivariant feature vectors to invariant arrays since one goal of the model is to predict potential energies, which are invariant to the rotations of atomic configurations. We also provide the proof of equivariance of the NewtonNet model in the Supplementary Information as well.

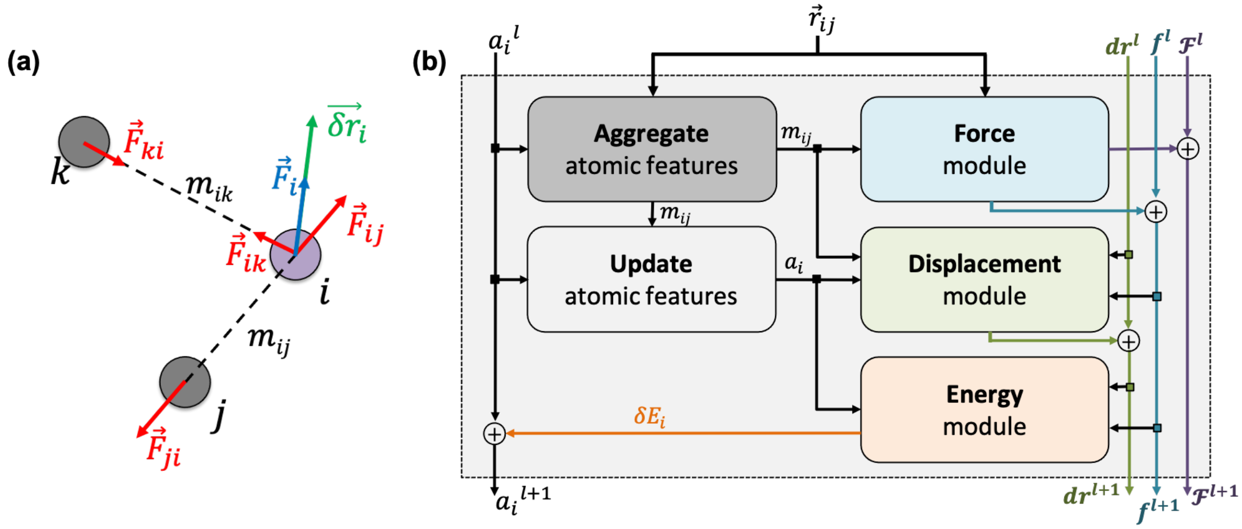

Figure 1: (a) Newton's laws for the force and displacement calculations for atom i with respect to its neighbors. (b) Schematic view of the NewtonNet message passing layer. At each layer four separate components are updated: atomic feature arrays a_i, latent force vectors $\boldsymbol{\mathcal{F}}$, and force and displacement feature vectors ($\boldsymbol{f}$ and $\boldsymbol{dr}$).

Atomic Feature Aggregator. We initialize the atomic features based on trainable embedding of atomic numbers Z_i, i.e., $a_i^0 = g(Z_i)$ and $g : \mathbb{R}^1 \to \mathbb{R}^{\text{nf}}$. We next use the edge function $e : \mathbb{R}^3 \to \mathbb{R}^{\text{nb}}$ to represent the interatomic distances using radial Bessel functions as introduced by Klicpera et al.[9]

$$e(\overrightarrow{r_{ij}}) = \sqrt{\frac{2}{r_c}} \frac{\sin(\frac{n\pi}{r_c} \|\overrightarrow{r_{ij}}\|)}{\|\overrightarrow{r_{ij}}\|} \tag{4}$$

where r_c is the cutoff radius and $\|\overrightarrow{r_{ij}}\|$ returns the interatomic distance between any atom i and j. We follow Schutt et al.[16] in using a self-interaction linear layer $\phi_{rbf} : \mathbb{R}^{\text{nb}} \to \mathbb{R}^{\text{nf}}$ to combine the output of radial basis functions with each other. This operation is followed by using an envelop function to implement a continuous radial cutoff around each atom. For this purpose, we use the polynomial function e_{cut} introduced by Klicpera et al.[9] with the choice of degree of polynomial $p = 7$. Thus, the edge operation $\phi_e : \mathbb{R}^3 \to \mathbb{R}^{\text{nf}}$ is defined as a trainable transformation of relative atom position vectors in the cutoff radius r_c

$$\phi_e(\overrightarrow{r_{ij}}) = \phi_{rbf}(e(\overrightarrow{r_{ij}}))\; e_{cut}(r_c, \|\overrightarrow{r_{ij}}\|)\,. \tag{5}$$

The output of ϕ_e is rotationally invariant as it only depends on the interatomic distances. Following the notation of neural message passing, we define a message function to collect the neighboring information and update atomic features. Here, we tend to pass a symmetric message between any pair of atoms, i.e., the message that is passed between atom i and atom j are the same in both directions. Thus, we introduce our symmetric message passing m_{ij} by element-wise product between all feature arrays involved in any two-body interaction,

$$m_{ij} = \phi_a(a_i^l)\ \phi_a(a_j^l)\ \phi_e(\overrightarrow{r_{ij}}) \tag{6}$$

where $\phi_a : \mathbb{R}^{\mathrm{nf}} \rightarrow \mathbb{R}^{\mathrm{nf}}$ indicates a trainable and differentiable network with a nonlinear activation function SiLU[19] after the first layer. Note that the ϕ_a is the same function applied to all atoms. Thus, due to the weight sharing and multiplication of output features of both heads of the two-body interaction, the m_{ij} remain symmetric at each layer of message passing. To complete the feature array aggregator, we use the equation 2 to simply sum all messages received by central atom i from its neighbors $\mathcal{N}(i)$. Finally, we update the atomic features at each layer using the sum of received messages,

$$a_i^{l+1} = a_i^l + \sum_{j \in \mathcal{N}(i)} m_{ij}. \tag{7}$$

Force Calculator. So far, we have followed a standard message passing that is invariant to the rotation. We begin to take advantage of directional information starting from the force calculator module. The core idea behind this module is to construct latent force vectors using the Newton's third law. The third law states that the force that atom i exerts on atom j is equal and in opposite direction of the force that atom j exerts on atom i. This is the reason that we intended to introduce a symmetric message passing operator. Thus, we can estimate the symmetric force magnitude as a function of m_{ij}, i.e., $\|\overrightarrow{F}_{ij}\| = \phi_F(m_{ij})$. The product of the force magnitude by unit distance vectors $\hat{r}_{ij} = \overrightarrow{r}_{ij}/\|\overrightarrow{r}_{ij}\|$ gives us antisymmetric interatomic forces that obey the Newton's third law (note that $\overrightarrow{r}_{ij} = -\overrightarrow{r}_{ji}$),

$$\overrightarrow{F}_{ij}^l = \phi_F(m_{ij})\, \hat{r}_{ij} \tag{8}$$

where $\phi_F : \mathbb{R}^{\mathrm{nf}} \rightarrow \mathbb{R}^1$ is a differentiable learned function, and $\overrightarrow{F}_{ij}^l \in \mathbb{R}^3$. The total force at each layer $\overrightarrow{F}_i^l$ on atom i is the sum of all the forces from the neighboring atoms j in the atomic environment,

$$\overrightarrow{F}_i^l = \sum_{j \in \mathcal{N}(i)} \overrightarrow{F}_{ij}^l, \tag{9}$$

and updating the latent force vectors at each layer,

$$\boldsymbol{\mathcal{F}}_i^{l+1} = \boldsymbol{\mathcal{F}}_i^l + \overrightarrow{F}_i^l. \tag{10}$$

We ultimately use the latent force vector from the last layer L, $\boldsymbol{\mathcal{F}}_i^L \in \mathbb{R}^3$ in the loss function to ensure this latent space truly mimics the underlying physical rules.

To complete the force calculator module, we borrow the idea of continuous filter from Schut et al. to decompose and scale latent force vectors along each dimension using another learned function $\phi_f : \mathbb{R}^{\mathrm{nf}} \rightarrow \mathbb{R}^{\mathrm{nf}}$. This way we can featurize the vector field to avoid too much of abstraction in the structural information that they carry with themselves,

$$\boldsymbol{\Delta f}_i = \sum_{j \in \mathcal{N}(i)} \phi_f(m_{ij})\, \vec{F}^{l}_{ij}. \tag{11}$$

As a result, the constructed latent interatomic forces are decomposed by rotationally invariant features along each dimmension, i.e., $\boldsymbol{\Delta f}_i \in \mathbb{R}^{3\times \mathrm{nf}}$. We call this type of representation feature vectors. Following the message passing strategy, we update the force feature vectors with $\boldsymbol{\Delta f}_i$ after each layer, while they are initialized with zero values, $\boldsymbol{f}_i^0 = \mathbf{0}$,

$$\boldsymbol{f}_i^{l+1} = \boldsymbol{f}_i^{l} + \boldsymbol{\Delta f}_i. \tag{12}$$

Momentum Calculator. This is the step that we try to estimate a measure of atomic displacement due to the forces that are exerted on them. We accumulate their dispalcements at each layer without updating the position of each atom. The main idea in this module is that the displacement must be along the updated force features in the previous step. Inspired by Newton's second law, we approximate the displacement factor using a learned function $\phi_r : \mathbb{R}^{\mathrm{nf}} \rightarrow \mathbb{R}^{\mathrm{nf}}$ that acts on the current state of each atom presented by its atomic features a_i^l,

$$\boldsymbol{\delta r}_i = \phi_r(a_i^{l+1})\boldsymbol{f_i^{l+1}}. \tag{13}$$

We finally update the displacement feature vectors by $\boldsymbol{\delta r}_i$ and a weighted sum of all the atomic displacements from the previous layer. The weights are estimated based on a trainable function of messages ($\phi_r^{'} : \mathbb{R}^{\mathrm{nf}} \rightarrow \mathbb{R}^{\mathrm{nf}}$) between atoms,

$$\boldsymbol{dr}_i^{l+1} = \sum_{j \in \mathcal{N}(i)} \phi_r^{'}(m_{ij})\, \boldsymbol{dr}_i^{l} + \boldsymbol{\delta r}_i. \tag{14}$$

The weight component in this step works like attention mechanism to concentrate on the two-body interactions that cause maximum movement in the atoms. Since forces at $l = 0$ are zero, the displacements are also initialized with zero values, i.e., $\boldsymbol{dr_i^0} = \mathbf{0}$.

Energy Calculator. The last module contracts the directional information to the rotationally invariant atomic features. Since we developed the previous steps based on the Newton's equations of motion, one immediate idea is to approximate the potential energy change for each atom using f_i^l and δr_i^l, resembling $\boldsymbol{f_i^l} \approx -\delta U/\boldsymbol{\delta r_i^l}$ in the higher dimensional space ($\mathbb{R}^{\mathrm{nf}}$). Thus, we find energy change for each atom by

$$\delta U_i = -\phi_u(a_i^{l+1}) \left\langle \boldsymbol{f}_i^{l+1} \cdot \boldsymbol{dr}_i^{l+1} \right\rangle, \tag{15}$$

where $\delta U_i \in \mathbb{R}^{\mathrm{nf}}$ and $\phi_u : \mathbb{R}^{\mathrm{nf}} \rightarrow \mathbb{R}^{\mathrm{nf}}$ is a differentiable learned function that operates on the atomic features and predicts the energy coefficient for each atom. The dot product of two feature vectors contracts the features along each dimension to a single feature array. We finally update the atomic features once again using the contracted directional information presented as atomic potential energy change,

$$a_i^{l+1} = a_i^{l+1} + \delta U_i. \tag{16}$$

This approach is both physically and mathematically consistent with the rotational equivariance operations and the goals of our model development. Physically, the energy change is the meaningful addition to the atomic feature arrays as they are used to predict the atomic energies eventually. Mathematically, the dot product of two feature vectors contracts the rotationally equivariant features to invariant features similar to euclidean distance that we used in the *atomic feature aggregator* module. Note that none of the force, displacement or energy modules are directly mapped to the final energy and force predictions. These are intermediate steps that update atomic features iteratively beyond the immediate neighborhood of each atom.

RESULTS

Here we show that the NewtonNet model is capable of predicting the energies and forces across a wide range of available chemical data sets, thereby covering much of the application space for which machine learning models are being developed and used by other research groups.

Single Small Molecules. We first evaluate the performance of NewtonNet on the data generated from molecular dynamics trajectories using Density Functional Theory (DFT)[20] for 9 small organic molecules from the MD17[21,22] and the revised MD17[23] benchmarks. Despite reported outliers in the calculated energies associated with the original version of the MD17 data, we still show it for completeness. For training NewtonNet, we select a data size of 950 for training, 50 for validation, and remaining data for test (more than 100k per molecule); this data split is more ambitious than that used by kernel methods such as sGDML[24] and FCHL19[23], and is supported by other emerging machine learning models that utilize equivariant operators, e.g., NequIP[13] and PaiNN[16]. Table 1 and Supplementary Table 1 shows the performance of NewtonNet for both energy and forces on the hold-out test set for both MD17 data sets, illustrating that it can outperform invariant deep learning models (e.g., SchNet[7], PhysNet[8], and DimeNet[9]) and even in some cases state-of-the-art equivariant models such as NequIP (using the original rank 1 version) and PaiNN.

On a similar task we train NewtonNet on the CCSD/CCSD(T) data reported for 5 small molecules[21,22]. The significance of this experiment is the gold standard of theory that is used to obtain the data, and addressing the ultimate goal to evaluate a machine learning model at high reference accuracy with an affordable number of training samples. In this benchmark data, the training and test splits are fixed at that provided by the authors of the MD17 data (i.e., 1000 training and 500 test data)[22]. In Table 2 we compare our results with NequIP and sGDML in which NewtonNet not only outperforms the best reported prediction performance for three of the five molecules, but it remains competitive within the range of uncertainties for the other two molecules, and is robustly improved compared to the opponent kernel methods. More recently NequIP has updated their results by considering higher tensor ranks ($l > 1$) and addition of translation equivariant features in the same preprint reference. We consider these additions very promising if the computational cost of more complex operators are justified. A benefit of NewtonNet is that it is computationally efficient and scalable relative to the newer equivariant methods that incorporate higher order tensors in the equivariant operators[13,14], while still retaining chemical accuracy ($<$ 0.5 kcal/mol).

Table 1: The performance of models in terms of mean absolute error (MAE) for the prediction of energies (kcal/mol) and forces (kcal/mol/Å) of molecules in the MD17 data sets. We report results by averaging over four random splits of the data to define standard deviations. Best results in the standard deviation range are marked in bold.

		SchNet	PhysNet	DimeNet	FCHL19	sGDML	NequIP	PaiNN	NewtonNet
							(l=1)		
Aspirin	energy	0.370	0.230	0.204	0.182	0.19	-	**0.159**	**0.168**± 0.019
	forces	1.35	0.605	0.499	0.478	0.68	**0.348**	0.371	**0.348**±0.014
Ethanol	energy	0.08	0.059	0.064	**0.054**	0.07	-	0.063	0.078±0.010
	forces	0.39	0.160	0.230	**0.136**	0.33	0.208	0.230	0.264±0.032
Malonaldehyde	energy	0.13	0.094	0.104	**0.081**	0.10	-	0.091	0.096±0.013
	forces	0.66	0.319	0.383	**0.245**	0.41	0.337	0.319	0.323±0.019
Naphthalene	energy	0.16	0.142	0.122	**0.117**	0.12	-	**0.117**	**0.118**±0.002
	forces	0.58	0.310	0.215	0.151	0.11	0.096	**0.083**	**0.084**±0.006
Salicylic Acid	energy	0.20	0.126	0.134	0.114	0.12	-	**0.114**	**0.115**±0.008
	forces	0.85	0.337	0.374	0.221	0.28	0.238	0.209	**0.197**±0.004
Toluene	energy	0.12	0.100	0.102	**0.098**	0.10	-	**0.097**	**0.094**±0.005
	forces	0.57	0.191	0.216	0.203	0.14	0.101	0.102	**0.088**±0.002
Uracil	energy	0.14	0.108	0.115	**0.104**	**0.11**	-	**0.104**	**0.107**±0.004
	forces	0.56	0.218	0.301	**0.105**	0.24	0.172	0.140	0.149±0.003
Azobenzene	energy	-	0.197	-	-	**0.092**	-	-	0.142±0.003
	forces	-	0.462	-	-	0.409	-	-	**0.138**±0.010
Paracetamol	energy	-	0.181	-	-	0.153	-	-	**0.135**±0.004
	forces	-	0.519	-	-	0.491	-	-	**0.263**±0.010

Small Molecules with large Chemical Variations. In a separate experiment to validate NewtonNet we trained it using the ANI-1 dataset to predict energies for a large and diverse set of 20 million conformations sampled from ∼ 58k small molecules with up to 8 heavy atoms.[25] The challenges in regards this dataset are three-fold: first, the molecular compositions and conformations are quite diverse, with the total number of atoms ranging from 2 to 26, and with total energies spanning a range of near 3×10^5 kcal/mol; second, only energy information is provided, so a well-trained network needs to extract information more efficiently from the dataset to outcompete data-intensive invariant models; finally, a machine learning model that performs well on such a diverse dataset is more transferable to unseen data, and will have a wider application domain.

We tested the performance of NewtonNet on the ANI-1 dataset following the protocol described in Ref.20, using 80% of the conformations from each molecule for training, 10% for validation and 10% for testing. In Table 3 we show that by utilizing only 10% (2M) samples of the original ANI-1 data, NewtonNet yields a MAE in energies of 0.65 kcal/mol, very near the standard definitions of chemical accuracy, and halving the error compared to ANNs using the full 20M ANI-1 dataset. Even with only 5% of the data (1M), we achieve an MAE of 0.85 kcal/mol on energies that exceeds the original performance of the ANN network trained with all data. Note that unlike the data experiments above, atomic forces are not reported with the ANI-1 data set. Although the NewtonNet model is trained without taking advantage of additional force information for the atomic environments, it clearly confirms that the directional information are generally a significant completion to the atomic feature representation regardless of the tensor order of the output properties.

Reducing Required Data on Active Learning Datasets. An active learning (AL) approach

Table 2: The performance of models in terms of mean absolute error (MAE) for the prediction of energies (kcal/mol) and forces (kcal/mol/Å) of molecules at CCSD or CCSD(T) accuracy. We randomly select 50 snapshots of the training data as the validation set and average the performance of NewtonNet over four random splits to find standard deviations. Best results in the standard deviation range are marked in bold.

		sGDML	NequIP (l=1)	NewtonNet
Aspirin	energy	0.158	-	**0.100** ± 0.007
	forces	0.761	**0.339**	**0.356** ± 0.019
Benzene	energy	**0.003**	-	**0.004** ± 0.001
	forces	0.039	0.018	**0.011** ± 0.001
Ethanol	energy	**0.050**	-	**0.049** ± 0.007
	forces	0.350	**0.217**	0.282 ± 0.032
Malonaldehyde	energy	0.248	-	**0.045** ± 0.004
	forces	0.369	0.369	**0.285** ± 0.038
Toluene	energy	0.030	-	**0.014** ± 0.001
	forces	0.210	0.101	**0.080** ± 0.005

has been suggested as a means to further improve ANN performance through better data sampling, reducing data requirements to 10-25% of the original ANI data set as reported by Smith and coworkers.[26]. We have tested NewtonNet on two datasets generated through an AL approach, and in Table 3 we show that we can make improved predictions on the active learning extended ANI-1X data set reported by Smith and co-worker[26] as well as active learning data generated for linear alkane pyrolysis[27]. For the ANI-1X dataset, we achieve better energy and force predictions on the test set with as low as 10% of all training data that was created through an active learning procedure. For the alkane pyrolysis dataset, we are able to achieve better force predictions on the test set when compared to the mean absolute error of forces on the training set of the original work, by utilizing as low as 30% of all training data.

The better performance of NewtonNet on these two datasets suggests that our model is capable of utilizing information more efficiently, even from dataset with limited sizes and concentrated information such as those created by AL. However in both cases the original AI models lack equivariant features, and as this will propagate into the AL sampling approach, it is therefore not a complete proof of optimality for NewtonNet. Instead training NewtonNet using an AL sampling approach would be required to fully take advantage of the improved ML model capabilities in addition to biasing the distribution of training date towards more difficult examples. In future work we hope to test whether further performance enhancements beyond that reported in Table 3 is realized once NewtonNet is combined with AL training.

Methane Combustion Reaction. The methane combustion reaction data[28] exerts a more challenging task due to the complex nature of reactive species that are often high in energy, transient, and far from equilibrium such as free radical intermediates. Such stress tests are important for driving *ab initio* molecular dynamics simulations in which even relatively low-run DFT functionals are notoriously time-consuming and limited to small system sizes. We utilized the dataset provided by Zeng et al.[28], which contains 578,731 snapshots for training. We trained NewtonNet on 100%, 10% and 1% of the data and evaluated the performance of NewtonNet on two hold-out test sets.

Table 3: The test performance of the NewtonNet model on small fractions of the original 20 million molecules ANI-1.[25] We also consider data sets derived from active learning including the ANI-1X[26] and alkane pyrolysis[27] data sets. The 3 data sets are reported in terms of mean absolute error (MAE) for energies (kcal/mol) and forces (kcal/mol/Å). The ANI-1X energy and force errors are reported as the performance of the NewtonNet model on the COMP6 benchmark only considering conformations of a given molecule within 100 kcal/mol energy range to compare with those reported by Smith et. al[26]. For the 10% training of the ANI-1X data, we randomly sampled 5000 frames from the remaining and complete ANI-1X data for test set. For the alkane pyrolysis dataset, we randomly sampled 7,100 frames from the 35,496 training frames to define the test set.

	ANI	NewtonNet	NewtonNet
training set size	20,000,000	2,000,000	1,000,000
energies	1.30	0.65	0.85
	ANI-1X	NewtonNet	
training set size	4,956,005	495,600	
energies	1.61	1.45	
forces	2.70	1.79	
	Alkane pyrolysis[a]	NewtonNet	NewtonNet
training set size	35,496	28,396	10,000
forces (train)	9.68	5.69	7.58
forces (test)	/	6.50	8.71

[a] The test set performance for the alkane pyrolysis reaction was not reported in Ref. 22, so we compared our test set performance with the training set performance in Ref. 22.

One test set is provided by the original authors, comprised of 13,315 snapshots generated using the same procedure as the training data and we refer to it as out-of-distribution (OOD) hold-out test set. The other set is 13,315 random samples of training data that we hold out as the final test set and we refer to it as in-distribution (ID) hold-out test set. The main reason for considering two test sets is the large energy and force distribution shifts that is found between the original training and test sets.

The prediction correlation plots for both energies and forces on the ID test set and OOD test set of NewtonNet trained with 100% data were provided in Supplementary Figure 1. Table 4 shows that when NewtonNet is trained on all available data, it drives down the ID test error in energies and forces significantly, even outperforming the reported training error of the original DeepMD model.[28] Even when utilizing as low as 1% of the training data, NewtonNet still has an MAE on the ID hold-out test set that is close to chemical accuracy on energy prediction. Even though a distribution shift was observed between the original train and test sets, NewtonNet still has competitive energy and force prediction accuracy on the out-of-distribution test dataset. Given the similar performance of NewtonNet on OOD test set with 100% and 10% of training data, we argue the comparison on the OOD test set is mainly influenced by the aforementioned distribution shift.

Hydrogen Combustion Reaction. This benchmark data is newly generated for this study and probes reactive pathways of hydrogen and oxygen atoms through the combustion reaction mechanism reported by Li et. al.[29], and analyzed with calculated intrinsic reaction coordinate (IRC) scans of 19 biomolecular sub-reactions from Bertels et. al.[30]. Excluding 3 reactions that

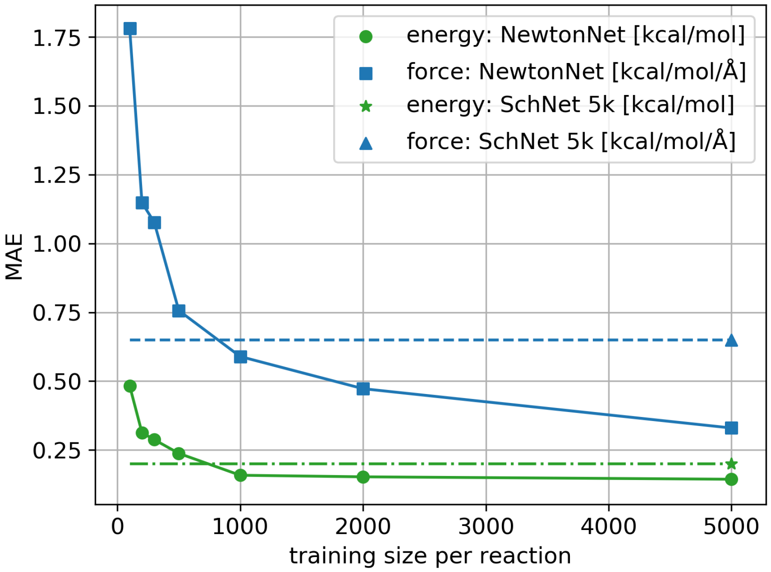

Figure 2: The learning curve of NewtonNet for the hydrogen combustion data, with MAEs of energy and forces averaged over the 16 independent reactions and with respect to the number of training samples used for each reaction. The dashed lines show the performance of SchNet when trained on all 5k data per sub-reaction.

Table 4: The performance of NewtonNet model compared with DeepMD on 13,315 randomly sampled in-distribution (ID) hold-out test configurations and 13,315 out-of-distribution (OOD) test configurations provided by the authors on the methane combustion dataset. Errors are reported in terms of mean absolute error (MAE) for energies (kcal/mol/atom) and forces (kcal/mol/Å). We systematically reduce the amount of training data by two orders of magnitude using NewtonNet and compare it to the 578,731 data points used in the original paper by Zeng and co-workers[28]

	DeepMD	NewtonNet	NewtonNet	NewtonNet
training set size	578,731	578,731	57,873	5,787
energies (ID)	0.945[a]	0.353	0.391	0.484
forces (ID)	/	1.12	1.88	2.78
energies (OOD)	3.227	3.170	3.135	3.273
forces (OOD)	2.77	2.75	2.93	3.76

[a] The MAE on the training set reported in ref. 14 was taken as the in-distribution prediction error here.

are chemically trivial (diatomic dissociation or recombination reactions), we obtain configurations and energies and forces for reactant, transition, and product states for 16 out of 19 reactions. The IRC data set was further augmented with normal mode displacements and AIMD simulations to sample configurations around the reaction path. All the calculations are conducted at the ωB97M-V/cc-pVTZ level of theory, and the data set comprises a total of ~280,000 potential energies and ~1,240,000 nuclear force vectors, and will be described in an upcoming publication.

We train NewtonNet on the complete reaction network by sampling training, validation, and test sets randomly formulated from the total data. The validation and test sizes are fixed to 1000 data per reaction, and the size of training data varies in a range of 100 to 5000 data points per reaction. The resulting model accuracy on the hold-out test set for both energy and forces is reported in Figure 2. It is seen that NewtonNet can outperform the best invariant SchNet model with slightly more than one order of magnitude smaller training data (500 vs 5000 samples per reaction), and is capable of achieving the chemical accuracy goal with as little as 500 data points per reaction. We integrate this model with the ASE interface[31] to run MD. A sample run is provided in Supplementary Figure 2 to demonstrate energy conservation. A more thorough study on this system using NewtonNet will come in later publications. In conventional deep learning approaches for reactive chemistry, abrupt changes in the force magnitudes can give rise to multimodal distributions of data, which can introduce covariate shift in the training of the models. Here we posit that a better representation of atomic environments using the latent force directions can increase the amount of attention that one atom gives to its immediate neighbors. As a result the performance of NewtonNet in prediction of forces for methane and hydrogen combustion reactive systems benefit most from the directional information provided by atoms that break or form new bonds.

DISCUSSION

Ablation Study. We justify our design choices for the NewtonNet architecture with ablation of network components on the example of the aspirin MD trajectory from the MD17 data set. Compared to the original design, we break the symmetric message passing in equation 6 by removing self atomic feature multiplication, and we investigate zeroing out the weight of latent force reconstruction loss in which the latent force vectors are not guided to the atomic forces direction. Note that in all these changes the number of model parameters remains constant.

Table 5 shows the results for various combination of these ablated components. The performance of the model deteriorates after each ablation, with the maximum change for breaking the symmetry and minimum change for removing the latent force reconstruction loss. The ablation of both together is also tested to confirm that even without the latent force loss, the entire design still needs to follow Newtonian rules (e.g., via the ablated symmetric message passing) to achieve its best performance. Based on our hyperparameter search, we have noticed that the weight of different loss components can significantly change the focus of the model on the energy or force optimization. We generally recommend a higher weight for force loss (λ_F) compared to other components. The weight of latent force loss (λ_D) can be even removed or faded out for some chemical systems with no or minimum change in the overall performance. However, breaking the second Newton law in our symmetric message passing function worsens the prediction performance significantly.

Table 5: Ablation study with a focus on the Newtonian components of our model. Numbers show the MAE of energy (kcal/mol) and force (kcal/mol/Å) predictions for aspirin molecule from MD17.

	energy	forces
no ablation	0.168 ± 0.019	0.348 ± 0.014
remove Sym. message passing	4.430 ± 2.020	4.290 ± 0.360
remove latent force loss	0.167 ± 0.014	0.359 ± 0.013
remove both	0.187 ± 0.022	0.427 ± 0.009

Computational Efficiency of NewtonNet. In addition to data efficiency as illustrated in

Results, NewtonNet allows for a linear scaling in computational complexity with respect to the number of atoms in the system. This can be mathematically proven since the value of all operators are proportional to the size of the system with the assumption that all neighbors are in a small cutoff radius. To give a better sense of the computational efficiency we compare the time that is needed to train on the aspirin molecule from the MD17 data set with the same calculation using the NequIP model. As reported by Batzner et. al., a complete training on the MD17 data to converge to the best performance of NequIP model takes up to 8 days.[13] However, NewtonNet only required 12 hours to give the state-of-the-art performance on a GeForce RTX 2080Ti, GPU which is only 73% as fast as the Tesla V100 that is used for evaluating NequIP, when a straightforward comparison of similar rank of contributed tensors used by both methods. Obviously, higher order tensors may boost performance but will increase the computation time, and should be analyzed from a cost-benefit perspective to find the best level of ML models for the required accuracy versus computational resources.

Aside from training time that is important to facilitate the model development and to reduce the testing time, the computation time per atomic environment is critical for the future application of trained models in an MD simulation. The computation time for processing a snapshot of the MD trajectory of a small molecule by NewtonNet is 4 milliseconds ($\sim$ 3 ms on a Tesla V100) for a small molecule of 20 atoms. Considering the reported average time of 16 milliseconds for NequIP to process a molecule of 15 atoms[13], NewtonNet demonstrates a significant speedup. In addition, the PaiNN model[16] is the closest to our model in terms of computational complexity, but does not encode additional physical knowledge in the message passing operators. As a result, it includes about 20% more number of optimized parameters (600k vs. 500k parameters in NewtonNet). This difference likely leads to higher computational cost with an equally efficient implementation of the code. Nevertheless, all these reported prediction times are by far smaller than *ab initio* calculations even for a snapshot of a small molecule in the MD trajectory, which is on the order of minutes to hours.

CONCLUSIONS

The ability to predict the energy and forces of a molecular dynamics trajectory with high accuracy but at an efficient time scale is of considerable importance in the study of chemical and biochemical systems. We have developed a new ML model based on Newton's equation of motion that can conduct this task more accurately (or achieve competitive performance) than other state-of-the-art invariant and equivariant models.

Overall, the presented results are promising in at least three major respects. First, since the NewtonNet model takes advantage of geometric message passing and a rotationally equivariant latent space which scales linearly with the size of the system, its promising performance in accuracy can be achieved without much computation or memory overhead. NewtonNet, like other equivariant deep learning models, utilizes less data and can still outperform the kernel methods that are renowned for their good performance on small size data. Given the better scalability of deep learning models such as NewtonNet compared to kernel methods, we can expand the training data, for example by smart sampling methods like active learning[32,33] and explore the potential energy surface of the chemical compound space more efficiently. The study of methane combustion reaction is a proof of evidence for this approach as the training data is a result of active learning sampling. If this sampling was initiated with NewtonNet predictions, one could achieve the best performance with even less number of queries.

Second, the data efficiency is the key to achieve ML force field models at the high accuracy levels of *first principles* methods such as CCSD(T)/CBS with competitive performances as state-of-the-art kernel-based methods using significantly less training data. For the CCSD(T) data on small single organic molecules we found that the NewtonNet performance is competitive or better than state-of-the-art equivariant models by at least 10%. This is a very encouraging result for being able to obtain gold standard levels of theory with affordable data set generation.

Finally, taking advantage of Newton's laws of motion in the design of the architecture helped to avoid unnecessary operations, and provide a more understandable and interpretable latent space to carry out the final predictions. Inspired by other physical operations that incorporate higher order tensors[13,14], NewtonNet can also be further extended to construct more distinguishable latent space many-body features in future work. Even so, the performance of the NewtonNet model on the MD trajectories from combustion reactions are both excellent with good chemical accuracy even when considering the challenge of chemical reactivity.

The idea to utilize directional information in neural networks in the form of equivariant operators is so recent that their broader application in chemical sciences are still in their early stage. One key feature in our experiments is the existence of atomic labels (i.e., forces) that help to propagate directional information smoothly and robustly. When outputs are only provided in a more abstract way (e.g, predicting molecular properties), the addition of domain knowledge in the form of regularizers or normalization layers[34] remain a challenge that domain researchers need to overcome to achieve the state of the art performance.

Appendix A. Proof of Equivariance and Invariance

We prove that our model is rotationally equivariant on the atomic positions $\mathbb{R}_i \in \mathbb{R}^3$ and atomic numbers Z_i for a rotation matrix $T \in \mathbb{R}^{3\times3}$. In the equation 1, the euclidean distance is invariant to the rotation, as it can be shown that

$$\begin{aligned}\left\|T\mathbf{r}_{ij}\right\|^2 &= \\ \left\|T\mathbf{R}_j - T\mathbf{R}_i\right\|^2 &= \\ \left(\mathbf{R}_j - \mathbf{R}_i\right)^\top T^\top T \left(\mathbf{R}_j - \mathbf{R}_i\right) &= \\ \left(\mathbf{R}_j - \mathbf{R}_i\right)^\top \mathbf{I} \left(\mathbf{R}_j - \mathbf{R}_i\right) &= \\ \left\|\mathbf{R}_j - \mathbf{R}_i\right\|^2 &= \\ \left\|\mathbf{r}_{ij}\right\|^2, &\end{aligned} \tag{17}$$

which means that the euclidean distance is indifferent to the rotation of the positions as it is quite well-known for this feature. Consequently, feature arrays m_{ij}, a_i, and all the linear or non-linear functions acting on them will result in invariant outputs. The only assumptions for this proof is that a linear combination of vectors or their product with invariant features will remain rotationally equivariant. Base on this assumption we claim that equation 5 to 11 will remain equivariant to the rotations. For instance, the same rotation matrix T propagates to equation 5 such that,

$$\phi_F(Tm_{ij})\,T\hat{r}_{ij} = \phi_F(m_{ij})\,T\hat{r}_{ij} = T\;\phi_F(m_{ij})\,\hat{r}_{ij} = T\overrightarrow{F}^{l}_{ij}. \tag{18}$$

The last operator, equation 12, will remain invariant to the rotations due to the use of dot product. The proof for the invariant atomic energy changes is that,

$$\begin{aligned}
-\phi_u(a_i^{l+1})\left(T\boldsymbol{f}_i^{l+1}\cdot T\boldsymbol{dr}_i^{l+1}\right) = \\
-\phi_u(a_i^{l+1})\left(\boldsymbol{f}_i^{l+1}\;T^{\top}T\;\boldsymbol{dr}_i^{l+1}\right) = \\
-\phi_u(a_i^{l+1})\left(\boldsymbol{f}_i^{l+1}\;\mathbf{I}\;\boldsymbol{dr}_i^{l+1}\right) = \\
-\phi_u(a_i^{l+1})\left(\boldsymbol{f}_i^{l+1}\cdot\boldsymbol{dr}_i^{l+1}\right) = \\
\delta U_i.
\end{aligned} \tag{19}$$

This is how we contract equivariant features to invariant arrays. The addition of these arrays to atomic features preserves the invariance for the final prediction of atomic contributions to the total potential energy.

METHODS

Training Details. We follow the summation rule as described by Behler and Parrinello[5] to predict the atomic energies. Following this rule, we use a differentiable function to map the updated atomic features after last layer a_i^L to atomic potential energies E_i. Ultimately, the total potential energy is predicted as the sum of all atomic energies.

$$E_i = \phi_{out}(a_i^L), \tag{20}$$

$$\widetilde{E} = \sum_i^{N_m} E_i, \tag{21}$$

where N_m is the total number of atoms, and $\phi_{out} : \mathbb{R}^{\mathrm{nf}} \to \mathbb{R}^1$ is a fully connected network with Sigmoid Linear Unit (SiLU) activation[19] after each layer except the last layer.

To accelerate training and achieve better generalizability, we applied two different data normalization approaches. For the dedicated small molecule models, we normalized the total potential energies $\widetilde{E}$ with fixed mean and standard deviation calculated from the training dataset. For methane combustion reaction models, we normalized the atomic potential energies E_i using trainable mean and standard deviation, and inverse normalize atomic energies before summing them up to allow variability in species compositions.

We obtain forces as gradient of potential energy with respect to atomic positions. This way we guarantee the energy conservation[23] and provide atomic forces for a robust training of the atomic environments,

$$\widetilde{\boldsymbol{F_i}} = -\nabla_i \widetilde{E}. \tag{22}$$

We train the model using small batches of data with batch size M. The loss function penalizes the model for predicted energy values, force components, and the direction of latent force vectors from last message passing layer $\boldsymbol{\mathcal{F}}_i^L$. These three terms of the loss function $\mathcal{L}$ are formulated as:

$$\mathcal{L} = \frac{\lambda_E}{M} \Sigma_m^M \left(\widetilde{E}_m - E_m \right)^2 + \\ \frac{\lambda_F}{M} \Sigma_m^M \frac{1}{3N_m} \sum_i^{N_m} \left\| \widetilde{\boldsymbol{F}}_{mi} - \boldsymbol{F}_{mi} \right\|^2 + \\ \frac{\lambda_D}{M \times N_m} \sum_m^M \sum_i^{N_m} \left(1 - \frac{\boldsymbol{\mathcal{F}}_{mi}^L \cdot \boldsymbol{F}_{mi}}{\left\| \boldsymbol{\mathcal{F}}_{mi}^L \right\| \left\| \boldsymbol{F}_{mi} \right\|} \right). \tag{23}$$

The first two terms are common choices for the energy and forces that are on the basis of the mean squared deviations of predicted values with references data. The last term penalizes the deviation of latent force vectors direction with the ground-truth force vectors. Here, we use cosine similarity loss function to minimize the $(1 - \cos(\alpha)) \in [0, 2]$, where α is the angle between the $\boldsymbol{\mathcal{F}}_i^L$ and $\boldsymbol{F_i}$ for each atom i of a snapshot m of a MD trajectory. The λ_E, λ_F, and λ_D are hyperparameters that determine the contribution of energy, force, and latent force direction losses in the total loss $\mathcal{L}$.

Table 6: Hyperparameters for all the reported experiments in the results section.

	λ_E	λ_F	λ_D	learning rate (lr)	lr decay	cutoff radius [Å]
MD17	1	50	1	1.10^{-3}	0.7	5
MD17/CCSD(T)	1	50	1	1.10^{-3}	0.7	5
ANI	1	0	0	1.10^{-4}	0.7	5
Methane combustion[a]	1	5	1	1.10^{-3}	0.7	5
Methane combustion[b]	1	5	1	1.10^{-4}	0.7	5
Hydrogen combustion	1	20	1	5.10^{-4}	0.7	5

[a] 10% data & 1% data. [b] 100% data.

We use mini-batch gradient descent algorithm (with Adam optimizer[35]) to minimize the loss function with respect to the trainable parameters. The trainable parameters are built in the learned functions noted with ϕ symbol. We use fully connected neural network with SiLU nonlinearity for all ϕ functions through out the message passing layer. The only exception is the ϕ_{rbf}, which is a single linear layer. We avoid using bias parameters in the ϕ_f and $\phi_r^{'}$ in order to propagate the radial cutoff throughout the network. We found it important for the ANI model to use a normalization layer[36] on the atomic features at every message passing layer as it helps with the stability of training. All NewtonNet models in this paper use $L = 3$ message passing layers, $nf = 128$ features and $nb = 20$ basis sets. The number of features are set similar to previous works to emphasize on the impact of architecture design in our comparisons. Other hyper-parameters are selected based on the best practices for each type of system and are reported in the Table 6.

For the training of SchNet in the hydrogen combustion study we use 128 features everywhere and 5 interaction layers as recommended by developers[7]. The other hyperparameters are the same as NewtonNet except for the force coefficient in the loss function that we found a lower $\lambda_F = 10$ performs better than larger coefficients.

ACKNOWLEDGMENTS

The authors thank the National Institutes of Health for support under Grant No 5U01GM121667 for support of the machine learning method. We also thank the National Science Foundation under grant CHE-1955643 for support of the chemical application of hydrogen combustion. FHZ thanks the Research Foundation-Flanders (FWO) Postdoctoral Fellowship for support as a visiting Berkeley scholar. M. Liu thanks the China Scholarship Council for a visiting scholar fellowship. C.J.S. acknowledges funding by the Ministry of Innovation, Science and Research of North Rhine-Westphalia ("NRW Rückkehrerprogramm") and an Early Postdoc.Mobility fellowship from the Swiss National Science Foundation. This research used computational resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

AUTHOR CONTRIBUTIONS

T.H-G. and M.H. conceived the scientific direction and wrote the complete manuscript. M.H., J.L., X.G., O.Z., M.L., and H.H. ran experiments with NewtonNet and other ML models on the various datasets and wrote the corresponding Results in the manuscript. A.D., C.J.S., L.B. and F.H-Z. generated the hydrogen combustion data. T.H-G., M.H., J.L., O.Z., F.H-Z., and X.G analyzed the results. All authors provided comments on the results and manuscript.

DATA AND CODE AVAILABILITY

The GitHub repository is publicly available and open source at https://github.com/THGLab/NewtonNet. We also designed a command line interface to facilitate faster implementation by non-programmers. Although the new hydrogen combustion data is currently under review it has been made available at https://github.com/THGLab/H2COMBUSTIONDATA.

DECLARATION OF INTERESTS

The authors declare no competing interests.

References

[1] Andrew L. Ferguson and Keith A. Brown. Data-driven design and autonomous experimentation in soft and biological materials engineering. *Annual Review of Chemical and Biomolecular Engineering*, 13(1):null, 2022. doi: 10.1146/annurev-chembioeng-092120-020803.

[2] Marina Garcia de Lomana, Fredrik Svensson, Andrea Volkamer, Miriam Mathea, and Johannes

Kirchmair. Consideration of predicted small-molecule metabolites in computational toxicology. *Digital Discovery*, pages –, 2022. doi: 10.1039/D1DD00018G.

[3] Ke Chen, Christian Kunkel, Karsten Reuter, and Johannes T. Margraf. Reorganization energies of flexible organic molecules as a challenging target for machine learning enhanced virtual screening. *Digital Discovery*, pages –, 2022. doi: 10.1039/D1DD00038A.

[4] Daniil Bash, Frederick Hubert Chenardy, Zekun Ren, Jayce J Cheng, Tonio Buonassisi, Ricardo Oliveira, Jatin N Kumar, and Kedar Hippalgaonkar. Accelerated automated screening of viscous graphene suspensions with various surfactants for optimal electrical conductivity. *Digital Discovery*, pages –, 2022. doi: 10.1039/D1DD00008J.

[5] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.*, 98:146401, Apr 2007. doi: 10.1103/PhysRevLett.98.146401.

[6] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural Message Passing for Quantum Chemistry. In *Proc. 34th Int. Conf. Mach. Learn. - Vol. 70*, ICML'17, pages 1263–1272. JMLR.org, 2017.

[7] K. T. Schütt, P. Kessel, M. Gastegger, K. A. Nicoli, A. Tkatchenko, and K.-R. Müller. Schnetpack: A deep learning toolbox for atomistic systems. *Journal of Chemical Theory and Computation*, 15(1):448–455, 2019. doi: 10.1021/acs.jctc.8b00908.

[8] Oliver T. Unke and Markus Meuwly. PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges. *J. Chem. Theory Comput.*, 15(6):3678–3693, 2019. doi: 10.1021/acs.jctc.9b00181.

[9] Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional Message Passing for Molecular Graphs. *arXiv preprint arXiv:2003.03123v1*, pages 1–13, 2020.

[10] Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch–gordan nets: a fully fourier space spherical convolutional neural network. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[11] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

[12] Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. *Adv. Neural Inf. Process. Syst.*, 32(NeurIPS), 2019.

[13] Simon Batzner, Tess E. Smidt, Lixin Sun, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, and Boris Kozinsky. Se(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *arXiv preprint arXiv:2101.03164*, 2021.

[14] Zhuoran Qiao, Anders S. Christensen, Matthew Welborn, Frederick R. Manby, Anima Anandkumar, and Thomas F. Miller. Unite: Unitary n-body tensor equivariant network with applications to quantum chemistry. *arXiv preprint arXiv:2105.14655*, 2021.

[15] Zachary L. Glick, Alexios Koutsoukas, Daniel L. Cheney, and C. David Sherrill. Cartesian message passing neural networks for directional properties: Fast and transferable atomic multipoles. *The Journal of Chemical Physics*, 154(22):224103, 2021. doi: 10.1063/5.0050444.

[16] Kristof T. Schütt, Oliver T. Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *arXiv preprint arXiv:2102.03150*, 2021.

[17] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model CNNs. *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, 2017-January: 5425–5434, 2017. doi: 10.1109/CVPR.2017.576.

[18] Raghunath O. Ramabhadran and Krishnan Raghavachari. Extrapolation to the gold-standard in quantum chemistry: Computationally efficient and accurate ccsd(t) energies for large molecules using an automated thermochemical hierarchy. *Journal of Chemical Theory and Computation*, 9(9):3986–3994, 2013. doi: 10.1021/ct400465q. URL `https://doi.org/10.1021/ct400465q`. PMID: 26592394.

[19] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107(2015):3–11, 2018. doi: 10.1016/j.neunet.2017.12.012.

[20] R.G. Parr and Y. Weitao. *Density-Functional Theory of Atoms and Molecules*. International Series of Monographs on Chemistry. Oxford University Press, 1994. ISBN 9780195357738.

[21] Stefan Chmiela, Alexandre Tkatchenko, Huziel E Sauceda, Igor Poltavsky, Kristof T Schütt, Klaus-robert Müller, Igor Poltavsky, and Kristof T Sch. Machine learning of accurate energy-conserving molecular force fields. *Sci. Adv.*, 3(5), 2017. doi: 10.1126/sciadv.1603015.

[22] Stefan Chmiela, Huziel E. Sauceda, Klaus Robert Müller, and Alexandre Tkatchenko. Towards exact molecular dynamics simulations with machine-learned force fields. *Nat. Commun.*, 9(1), 2018. doi: 10.1038/s41467-018-06169-2.

[23] Anders S. Christensen, Lars A. Bratholm, Felix A. Faber, and O. Anatole von Lilienfeld. Fchl revisited: Faster and more accurate quantum machine learning. *The Journal of Chemical Physics*, 152(4):044107, 2020. doi: 10.1063/1.5126701.

[24] Stefan Chmiela, Huziel E. Sauceda, Igor Poltavsky, Klaus-Robert Müller, and Alexandre Tkatchenko. sgdml: Constructing accurate and data efficient molecular force fields using machine learning. *Computer Physics Communications*, 240:38–45, 2019. doi: https://doi.org/10.1016/j.cpc.2019.02.007.

[25] Justin S Smith, Olexandr Isayev, and Adrian E Roitberg. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chemical science*, 8(4):3192–3203, 2017.

[26] Justin S. Smith, Ben Nebgen, Nicholas Lubbers, Olexandr Isayev, and Adrian E. Roitberg. Less is more: Sampling chemical space with active learning. *The Journal of Chemical Physics*, 148(24):241733, 2018. doi: 10.1063/1.5023802.

[27] Jinzhe Zeng, Linfeng Zhang, Han Wang, and Tong Zhu. Exploring the chemical space of linear alkane pyrolysis via deep potential generator. *Energy & Fuels*, 35(1):762–769, 2021. doi: 10.1021/acs.energyfuels.0c03211. URL `https://doi.org/10.1021/acs.energyfuels.0c03211`.

[28] Jinzhe Zeng, Liqun Cao, Mingyuan Xu, Tong Zhu, and John ZH Zhang. Complex reaction processes in combustion unraveled by neural network-based molecular dynamics simulation. *Nature communications*, 11(1):1–9, 2020.

[29] Juan Li, Zhenwei Zhao, Andrei Kazakov, and Frederick L. Dryer. An updated comprehensive kinetic model of hydrogen combustion. *Int. J. Chem. Kinet.*, 36(10):566–575, 2004. doi: 10.1002/kin.20026.

[30] Luke W. Bertels, Lucas B. Newcomb, Mohammad Alaghemandi, Jason R. Green, and Martin Head-Gordon. Benchmarking the Performance of the ReaxFF Reactive Force Field on Hydrogen Combustion Systems. *J. Phys. Chem. A*, 124(27):5631–5645, 2020. ISSN 15205215. doi: 10.1021/acs.jpca.0c02734.

[31] Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N Groves, Bjørk Hammer, Cory Hargus, Eric D Hermes, Paul C Jennings, Peter Bjerre Jensen, James Kermode, John R Kitchin, Esben Leonhard Kolsbjerg, Joseph Kubal, Kristen Kaasbjerg, Steen Lysgaard, Jón Bergmann Maronsson, Tristan Maxson, Thomas Olsen, Lars Pastewka, Andrew Peterson, Carsten Rostgaard, Jakob Schiøtz, Ole Schütt, Mikkel Strange, Kristian S Thygesen, Tejs Vegge, Lasse Vilhelmsen, Michael Walter, Zhenhua Zeng, and Karsten W Jacobsen. The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, 29 (27):273002, 2017. URL `http://stacks.iop.org/0953-8984/29/i=27/a=273002`.

[32] Mojtaba Haghighatlari, Gaurav Vishwakarma, Doaa Altarawy, Ramachandran Subramanian, Bhargava U. Kota, Aditya Sonpal, Srirangaraj Setlur, and Johannes Hachmann. Chemml: A machine learning and informatics program package for the analysis, mining, and modeling of chemical and materials data. *WIREs Computational Molecular Science*, 10(4):e1458, 2020. doi: https://doi.org/10.1002/wcms.1458.

[33] April Mae Miksch, Tobias Morawietz, Johannes Kästner, Alexander Urban, and Nongnuch Artrith. Strategies for the Construction of Machine-Learning Potentials for Accurate and Efficient Atomic-Scale Simulations. *Mach. Learn. Sci. Technol.*, 2021. doi: 10.1088/2632-2153/abfd96.

[34] Mojtaba Haghighatlari, Gaurav Vishwakarma, Mohammad Atif Faiz Afzal, and Johannes Hachmann. A Physics-Infused Deep Learning Model for the Prediction of Refractive Indices and Its Use for the Large-Scale Screening of Organic Compound Space. *ChemRxiv*, pages 1–9, 2019.

[35] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pages 1–15, 2015.

[36] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer norm. *arXiv preprint arXiv:1607.06450v1*, 2015.