

RESEARCH ARTICLE | JUNE 09 2021

Cartesian message passing neural networks for directional properties: Fast and transferable atomic multipoles

Zachary L. Glick ; Alexios Koutsoukas; Daniel L. Cheney; C. David Sherrill  

 Check for updates

J. Chem. Phys., 154, 224103 (2021)
<https://doi.org/10.1063/5.0050444>

 CHORUS



View
Online



Export
Citation

Articles You May Be Interested In

An intelligence model to forecast blood brain barrier permeability using graph neural network

AIP Conf. Proc. (June 2024)

Intelligent performance inference: A graph neural network approach to modeling maximum achievable throughput in optical networks

APL Mach. Learn. (May 2023)

Deep learning for energetic material detonation performance

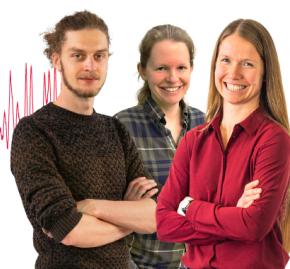
AIP Conf. Proc. (November 2020)

Webinar From Noise to Knowledge

May 13th – Register now



Universität
Konstanz



Cartesian message passing neural networks for directional properties: Fast and transferable atomic multipoles

Cite as: J. Chem. Phys. 154, 224103 (2021); doi: 10.1063/5.0050444

Submitted: 15 March 2021 • Accepted: 20 May 2021 •

Published Online: 9 June 2021



View Online



Export Citation



CrossMark

Zachary L. Glick,¹ Alexios Koutsoukas,² Daniel L. Cheney,² and C. David Sherrill^{1,a)}

AFFILIATIONS

¹ Center for Computational Molecular Science and Technology, School of Chemistry and Biochemistry, and School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332-0400, USA

² Molecular Structure and Design, Bristol Myers Squibb Company, P.O. Box 5400, Princeton, New Jersey 08543, USA

^{a)}Author to whom correspondence should be addressed: sherrill@gatech.edu

ABSTRACT

The message passing neural network (MPNN) framework is a promising tool for modeling atomic properties but is, until recently, incompatible with directional properties, such as Cartesian tensors. We propose a modified Cartesian MPNN (CMPNN) suitable for predicting atom-centered multipoles, an essential component of *ab initio* force fields. The efficacy of this model is demonstrated on a newly developed dataset consisting of 46 623 chemical structures and corresponding high-quality atomic multipoles, which was deposited into the publicly available Molecular Sciences Software Institute QCArchive server. We show that the CMPNN accurately predicts atom-centered charges, dipoles, and quadrupoles and that errors in the predicted atomic multipoles have a negligible effect on multipole–multipole electrostatic energies. The CMPNN is accurate enough to model conformational dependencies of a molecule’s electronic structure. This opens up the possibility of recomputing atomic multipoles on the fly throughout a simulation in which they might exhibit strong conformational dependence.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0050444>

11 August 2025 11:36:56

I. INTRODUCTION

Although the many-body Schrödinger equation can be solved, in principle, for any system, in many situations, the accuracy of *ab initio* methods does not justify their steep computational cost. Efficient, large-scale simulations of chemical systems necessitate inexpensive force fields, such as AMBER,¹ CHARMM,² or OPLS.³ These classical force fields contain simple functional forms and are parameterized to reproduce a combination of experimental and computational results. By contrast, *ab initio* force fields, such as the effective fragment potential (EFP),⁴ Gaussian electrostatic model (GEM),⁵ and sum of interactions between fragments *ab initio* (SIBFA),⁶ offer a bridge between the rigor of quantum mechanics and the convenience of molecular mechanics.⁷ The functional forms of *ab initio* force fields are generally inspired by intermolecular perturbation theory and consequently contain few, if any, empirically determined parameters. Instead, parameters are obtained directly from or else fitted to reproduce *ab initio* computations.

There are a couple of limitations to this class of force fields. For one, the required *ab initio* calculation may be non-trivial, especially when studying large macromolecules. To address this concern, the biomolecular EFP (bioEFP) method was recently developed to study large polypeptide systems by utilizing *ab initio* calculations on capped peptide fragments.⁸ An additional concern is that using *static* molecule-specific parameters ignores conformational dependence of the molecule’s electronic structure. Local torsional effects in flexible molecules have been shown to influence charge distributions.^{9,10} Force fields that use fixed parameters are unable to capture these effects.

Advances in machine learning (ML) are capable of addressing both these limitations. Using ML to predict atomic properties, often with neural networks (NNs), has gradually gained traction over the past decade, from the first attempts to model atomic densities of a single molecule by Darley, Handley, and Popelier¹¹ to the recently popularized graph-based NNs, such as AimNet, which model multiple atomic and molecular properties for entire classes of molecules.¹²

The common thread shared by all of these works is that the expensive quantum calculation is replaced with a prediction by a model trained on many previous quantum calculations, potentially at a very high level of theory. Like a force field, the computational cost of the model inference can be made asymptotically linear in system size, potentially allowing the force field parameters to be updated on the fly, similar to the learn on the fly (LOTF) method, which augments a simple parameterized force model with more expensive but more accurate calculations.¹³

Various properties can be modeled using this strategy, such as charge and atomic volume. In the case of ML potentials, one or more components of the force field are replaced by a ML model.^{12,14–27} Often, long-range interactions are modeled with a traditional force field, while a ML potential is used to model short range interactions, which are less likely to obey simple functional forms. One critical set of properties has been somewhat overlooked: the atomic electrostatic multipole expansion, which is necessary for a quantitative description of electrostatics and is important in any force field that accounts for atomic anisotropy.^{28,29}

With some notable exceptions, ML has been rarely used to develop transferable atomic multipole expansion models. While it may be tempting to attribute the scarcity of ML multipole models to a lack of interest in multipoles (relative to atomic charges and energies), a contributing factor is that higher order multipoles are much harder to model with standard ML techniques. This stems from the fact that higher order multipoles have a direction, making them dependent on a coordinate system. While an atomic charge is the same no matter how a molecule is rotated (invariant to rotation), an atomic dipole vector rotates with the molecule (equivariant to rotation). Until recently, ML models used in chemistry have been almost exclusively designed under the assumption of property invariance, making them unable to account for equivariance and therefore unable to model dipoles, quadrupoles, etc.

Early ML models of directional properties relied on using local reference frames and rotation invariant features, such as the single-molecule NNs of Darley, Handley, and Popelier¹¹ or the transferable kernel ridge regression (KRR) models of Bereau, Andrienko, and von Lilienfeld,³⁰ which were later refined in the IPML potential.²² The covariant kernel of Glielmo, Sollich, and De Vita³¹ was one of the first truly equivariant models of chemical systems and was used to predict atomic forces via Gaussian process regression (GPR). The subsequent symmetry-adapted GPR of Grisafi *et al.*³² generalized the covariant kernel to tensorial properties, in general, and has been used to model atomic dipoles³³ and even molecular polarizabilities.³⁴ Similarly, progress has been made in developing NNs to handle equivariance, such as the tensor field network³⁵ and the Cormorant network.³⁶ Miller *et al.*³⁷ performed a thorough ablation study to demonstrate the utility of using internally equivariant NN layers to predict molecular properties. The message passing neural network (MPNN) framework yields state-of-the-art regression results for atomic property prediction but has traditionally been limited to the prediction of rotationally invariant properties, such as energy and atomic charges.^{12,20} Very recently, MPNNs have been adapted to both use equivariant layers^{38,39} and generate accurate molecular spectra from molecular dynamics (MD) simulations.⁴⁰

Here, we extend the MPNN framework to handle the prediction of directional atomic properties and introduce the Cartesian MPNN

(CMPNN). The CMPNN uses rotationally equivariant features to predict Cartesian tensors. We then demonstrate that an implementation of this more flexible MPNN model does indeed predict higher order atomic multipoles with acceptable accuracy, speed, and transferability. We train a multitask CMPNN that simultaneously predicts atomic monopoles, dipoles, and quadrupoles on molecules relevant to protein/ligand interactions. While the higher order multipoles predicted by the CMPNN are not rigorously equivariant to all rotations, we demonstrate that by using an inexpensive form of data augmentation, the rotational variance of predicted multipoles is negligible. We use the MPNN-predicted multipoles to estimate the electrostatic interaction energy across the interaction coordinate of the doubly hydrogen bonded formamide dimer, and we show that the errors arising from ML are small relative to the errors in the multipole approximation itself.

II. THEORY

A. Multipole expansion

The electron density as computed with an *ab initio* method is usually expressed in a basis too large and expensive for practical use in a force field. Instead, the electrostatic potential generated by a molecule's charge distribution can be described with a multipole expansion, which is a valid approximation at long range. We use the notation of Stone²⁸ for the multipole expansion, the first three terms of which are the monopole or charge (q), dipole vector (μ), and traceless Cartesian quadrupole tensor (Θ),

$$q = \int d^3 \mathbf{r} \rho(\mathbf{r}), \quad (1)$$

$$\mu = \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix}, \quad (2)$$

$$\mu_\alpha = \int d^3 \mathbf{r} \rho(\mathbf{r}) \mathbf{r}_\alpha, \quad (3)$$

$$\Theta = \frac{3}{2} \begin{bmatrix} \Theta_{xx} & \Theta_{xy} & \Theta_{xz} \\ \Theta_{xy} & \Theta_{yy} & \Theta_{yz} \\ \Theta_{xz} & \Theta_{yz} & \Theta_{zz} \end{bmatrix}, \quad (4)$$

$$\Theta_{\alpha\beta} = \int d^3 \mathbf{r} \rho(\mathbf{r}) \left(\mathbf{r}_\alpha \mathbf{r}_\beta - \frac{1}{3} |\mathbf{r}|^2 \delta_{\alpha\beta} \right). \quad (5)$$

Although any charge distribution can be represented with a multipole expansion at a central point, a singular multipole expansion provides a poor description of an entire molecular electrostatic potential, except at very far distances from the expansion site. In order to accurately describe the electrostatic potential near the molecular surface—a requirement of intermolecular force fields—a set of distributed multipoles, often placed at atom centers, is necessary. Given a set of fixed sites for a distributed multipole expansion, many methods exist for extracting multipoles from an *ab initio* density, such as the Gaussian Distributed Multipole Analysis (GDMA) techniques of Stone.^{41,42} In this work, we use the minimal basis iterative stockholder (MBIS) charge partitioning scheme,⁴³ which is one

of the many Hirshfeld-like⁴⁴ atoms-in-molecules (AIM) methods,⁴⁵ to partition the molecular density into atomic densities, which can then be represented with a multipole expansion. The atomic multipoles produced by MBIS have been shown to accurately reproduce electrostatic interaction energies. Although we use MBIS for reference atomic multipoles, the CMPNN model proposed here could be trained to predict atomic multipoles from any partitioning scheme.

B. Message passing neural networks (MPNNs)

Neural networks (NNs) have emerged as one of the most versatile ML tools for modeling and predicting chemical properties. Early models required hand-designed features to represent atoms in molecules.¹⁴ Today, the state-of-the-art NNs are designed to effectively “learn” the optimal features by operating on a graph-based representation of entire molecular structures. These models, referred to interchangeably as either message passing neural networks (MPNNs) or graph convolutional networks (GCNs), have been applied to a wide array of problems across chemistry.^{17,20,46} Here, the MPNN terminology as defined by Gilmer *et al.*²⁰ is briefly reviewed so that we can extend it to the prediction of directional atomic properties, such as the atomic dipole and quadrupole. The outline of a generic MPNN is summarized in Fig. 1.

Within an MPNN, a molecule with N atoms is represented by a graph \mathcal{G} of N vertices and up to $N(N - 1)/2$ edges, where each vertex represents an atom and each edge represents a pair of nearby atoms. At a given vertex, the respective atom’s type is encoded in a vector

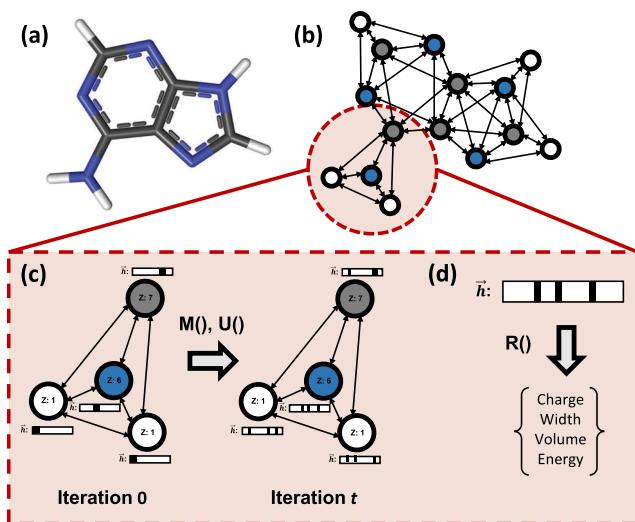


FIG. 1. Schematic of a standard MPNN used to predict atomic properties. (a) The MPNN model operates on an entire molecule, defined by a set of coordinates. (b) In the model, the molecule is represented as a graph, with vertices denoting atoms and edges encoding interatomic distances. (c) Vertices are initialized with state vectors (\bar{h}) that initially only encode the atom type. After t message passing iterations, in which vertices send and receive messages from neighboring vertices, each updated state vector encodes the unique atomic environment of the respective atom. (d) This final state vector is used to predict a desired atomic property.

\mathbf{h}_v^0 , called the initial hidden state vector. Similarly, interatomic distances are encoded in edge vectors (\mathbf{e}_{vw}). The MPNN is a function f that maps from \mathcal{G} to some atomic property \mathbf{y}_v ,

$$f : (\{\mathbf{h}_v^0\}, \{\mathbf{e}_{vw}\}) \rightarrow \{\mathbf{y}_v\}. \quad (6)$$

As indicated by its name, the MPNN is an iterative function, running for T message passing iterations or time steps. Conceptually, at every iteration, the hidden state vectors of the vertices are refined to better describe the associated atom’s environment. Initially, every vertex v is represented by the relatively uninformative hidden state \mathbf{h}_v^0 . At each time step $t \in [0, \dots, T - 1]$, the hidden state of v is updated from \mathbf{h}_v^t to \mathbf{h}_v^{t+1} based on a message \mathbf{m}_v^{t+1} sent to vertex v from the other nearby vertices. To do this, an MPNN must implement a message function (M_t) and update function (U_t),

$$\mathbf{m}_v^{t+1} = \sum_{w \in N(v)} M_t(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}), \quad (7)$$

$$\mathbf{h}_v^{t+1} = U_t(\mathbf{h}_v^t, \mathbf{m}_v^{t+1}). \quad (8)$$

M_t produces a message for a vertex v from each of its neighbor vertices $w \in N(v)$. The sum of all messages received by vertex v , \mathbf{m}_v^{t+1} , is used by the update function U_t to update the hidden state \mathbf{h}_v^t into \mathbf{h}_v^{t+1} . The superscript t indicates that the message and update functions may be iteration dependent. Finally, every MPNN implements a readout function (R) that is used to predict atomic properties from the final hidden state,

$$\mathbf{y}_v = R(\mathbf{h}_v^T). \quad (9)$$

We note that this definition of the readout function from Gilmer *et al.*²⁰ can be expanded to use all of the hidden states of the vertex, not just the final hidden state,

$$\mathbf{y}_v = R(\mathbf{h}_v^0, \dots, \mathbf{h}_v^t, \dots, \mathbf{h}_v^T). \quad (10)$$

The utility of the MPNN framework is a result of the intentionally abstract way in which the functions M_t , U_t , and R are invoked. These functions may be concatenations, dot products, matrix multiplications, neural networks, or any combination of these and other operations.

C. Rotational invariance and equivariance

We denote a rigid molecular rotation as \hat{R} . The predictions of the MPNN, as described in Sec. II B, are invariant to the rotation of the molecule,

$$f(\hat{R}\mathcal{G}) = \{\mathbf{y}_v\} \quad \forall \hat{R}. \quad (11)$$

Invariance occurs because the molecular graph \mathcal{G} is itself invariant to rotation, containing only Cartesian distances between atoms. This is the desired behavior for modeling many chemical properties, such as energy or charge. It is useful to consider how the same MPNN would fare at modeling a directional property, such as the atomic dipole vector μ_v , perhaps by predicting the three vector components ($\mu_{v,x}$, $\mu_{v,y}$, and $\mu_{v,z}$) for all v . Directional properties require rotational

equivariance,

$$f(\hat{R}\mathcal{G}) = \{\hat{R}\mathbf{y}_v\} \quad \forall \hat{R}. \quad (12)$$

The rotational invariance of the MPNN is clearly incompatible with the rotational equivariance of directional properties, such as dipoles.

One workaround to this problem might be to instead use the MPNN to predict $|\boldsymbol{\mu}_v|$, which is invariant to rotation. However, the dipole norm is not particularly useful without the associated direction. Alternatively, one could rotate every molecule (or every atomic environment) prior to applying the MPNN using some system of rules, e.g., closest atom defines the positive x axis, etc. This kind of local axis frame is sensible for modeling small symmetric molecules, but it is not clear that a set of rules exists that would be appropriate for a general and transferable model of directional atomic properties.

D. Atomic dipole MPNN

Modeling directional atomic properties with an MPNN requires modification of the molecular representation. We will introduce directional information by way of the edge features in \mathcal{G} . The edge feature vector \mathbf{e}_{vw} as defined in Sec. II B is an encoding of the scalar distance between nuclei v and w . This scalar distance does not encode the direction of the vector \mathbf{r}_{vw} , only the magnitude, making it the source of directional invariance. To address this problem, we will define a tuple of three Cartesian edge feature vectors ($\mathbf{e}_{vw,x}, \mathbf{e}_{vw,y}, \mathbf{e}_{vw,z}$) as

$$\mathbf{e}_{vw,\alpha} = \mathbf{e}_{vw} \frac{\mathbf{r}_{vw} \cdot \hat{\alpha}}{|\mathbf{r}_{vw}|}, \quad (13)$$

where $\hat{\alpha}$ is the unit vector along the positive α -axis. The Cartesian edge feature vectors $\mathbf{e}_{vw,\alpha}$ differ from the original feature vector \mathbf{e}_{vw} by a scaling factor that is a function of angle between \mathbf{r} and $\hat{\alpha}$. We note that this scaling factor gives the collection of vectors ($\mathbf{e}_{vw,x}, \mathbf{e}_{vw,y}, \mathbf{e}_{vw,z}$) the same equivariance to rotation as $\boldsymbol{\mu}_v$,

$$\hat{R}(\mathbf{e}_{vw,x}, \mathbf{e}_{vw,y}, \mathbf{e}_{vw,z}) = \hat{R}\boldsymbol{\mu}_v. \quad (14)$$

For example, if we rotate a molecule about the x axis, neither $\mathbf{e}_{vw,x}$ nor $\boldsymbol{\mu}_{v,x}$ change. A 180° rotation about the x axis flips the signs of $\mathbf{e}_{vw,y}$ and $\mathbf{e}_{vw,z}$ as well as $\boldsymbol{\mu}_{v,y}$ and $\boldsymbol{\mu}_{v,z}$. This makes $\mathbf{e}_{vw,\alpha}$ an excellent candidate for an edge feature to use in the CMPNN.

In many applications, one wants to predict a combination of rotation invariant and directional atomic properties (e.g., atomic monopole and dipole). Instead of implementing separate models to perform these two types of predictions, we can develop a multi-target model that performs both simultaneously.⁴⁷ The model is an extended version of the MPNN described earlier using the directional edge features ($\{\mathbf{e}_{vw,\alpha}\}$) in addition to the original features. For the directional task, we invoke a separate set of directional message, update, and readout functions (M_t^u, U_t^u, R^u) that are used to determine directional hidden state vectors ($\{\mathbf{h}_{v,\alpha}^t\}$),

$$\mathbf{m}_{v,\alpha}^{t+1} = \sum_{w \in N(v)} M_t^u(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw,\alpha}), \quad (15)$$

$$\mathbf{h}_{v,\alpha}^{t+1} = U_t^u(\mathbf{h}_v^t, \mathbf{m}_{v,\alpha}^{t+1}), \quad (16)$$

$$\boldsymbol{\mu}_{v,\alpha} = R^u(\mathbf{h}_{v,\alpha}^1, \dots, \mathbf{h}_{v,\alpha}^t, \dots, \mathbf{h}_{v,\alpha}^T). \quad (17)$$

These dipole functions have the same functional forms as their rotation-invariant counterparts, but their weights are optimized for the task of reproducing atomic dipole vector components instead of rotation-invariant atomic properties. Note that all three Cartesian dipole components are predicted by the same message, update, and readout functions. In addition, the rotation-invariant hidden state vector \mathbf{h}_v^t is present in the directional message function, allowing the prediction of both monopoles and dipoles to benefit from a shared internal representation.

E. Atomic quadrupole MPNN

A number of force fields, such as AMOEBA⁴⁸ and MASTIFF,⁴⁹ include the atomic quadrupole moment to describe electrostatic interactions. We can further extend the multitask CMPNN to this additional task. To model the quadrupole, we will have six unique edge vectors corresponding to the six components that define the Cartesian quadrupole moment: $\Theta_{xx}, \Theta_{xy}, \Theta_{xz}, \Theta_{yy}, \Theta_{yz}$, and Θ_{zz} ,

$$\mathbf{e}_{vw,\alpha\beta} = \mathbf{e}_{vw} \frac{(\mathbf{r}_{vw} \cdot \hat{\alpha})(\mathbf{r}_{vw} \cdot \hat{\beta})}{|\mathbf{r}_{vw}|^2}. \quad (18)$$

As in the dipole case, this feature vector has the same equivariance to rotation as the target property. An additional set of message, update, and readout functions is also needed,

$$\mathbf{m}_{v,\alpha\beta}^{t+1} = \sum_{w \in N(v)} M_t^\Theta(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw,\alpha\beta}), \quad (19)$$

$$\mathbf{h}_{v,\alpha\beta}^{t+1} = U_t^\Theta(\mathbf{h}_{v,\alpha\beta}^t, \mathbf{m}_{v,\alpha\beta}^{t+1}), \quad (20)$$

$$\mathbf{y}_v = R^\Theta(\mathbf{h}_{v,\alpha\beta}^1, \dots, \mathbf{h}_{v,\alpha\beta}^t, \dots, \mathbf{h}_{v,\alpha\beta}^T), \quad (21)$$

which again have the same functional forms as the monopole and dipole functions but have weights optimized for the task of predicting quadrupole components.

Generalizing this model to predict multipoles of any order would be straightforward. However, atomic multipoles beyond the quadrupole are rarely used in force fields, so the multitask CMPNN model described in this work will be limited to monopole, dipole, and quadrupole predictions.

III. METHODOLOGY

A. Training and testing data

For developing and testing the CMPNN, a dataset of unique, neutral molecular fragments was created using the following procedure. Chemical structures were obtained from version 25 of the ChEMBL database⁵⁰ and subsequently filtered by employing Lipinski's rule of five to retain drug-like molecules.^{51,52} These molecules were fragmented using in-house software, and a structurally diverse subset was chosen in which molecules were limited to an atom count of 40 and molecular weight of 250 and comprised only of the atomic elements most commonly encountered in drug discovery: H, C, N, O, F, P, S, Cl, and Br. The molecules were subjected to energy minimization with MacroModel using the OPLS3e

force field and the GBSA implicit solvent model.⁵³ This new dataset of 39 912 molecules was additionally augmented with QM7,^{54,55} which contains 7165 small organic molecules, resulting in a final count of 46 623 molecules after removal of molecules occurring in both sets. Figure 2 shows the distribution of molecular weights and a sample of structures in this dataset. Additional details regarding dataset generation, curation, and characterization can be found in the [supplementary material](#).

Reference electron densities for each molecule in the dataset were calculated with a development version of the Psi4 program⁵⁶ at the PBE0/aug-cc-pV(T+d)Z level of theory. The PBE0 functional has been shown to reliably approximate the coupled-cluster singles and doubles (CCSD) method for molecular electron density calculations.⁵⁷ Next, molecular densities were partitioned into atomic electron densities with the MBIS⁴³ routine of the Horton program.⁵⁸ These atomic densities are described with a multipole expansion, and only the first three terms are retained: the monopole, dipole, and quadrupole. In addition to the PBE0/aug-cc-pV(T+d)Z multipoles, we also calculated atomic multipoles at the less accurate HF/jun-cc-pV(D+d)Z level of theory in order to easily compare electrostatic interactions computed with atom-centered multipoles to electrostatic interactions computed with intermolecular perturbation theory. Unless otherwise stated, all reference MBIS multipoles and CMPNN predictions are at the PBE0/aug-cc-pV(T+d)Z level of theory. Throughout this work, a 80/10/10% split of the dataset is used for training, validating, and testing the MPNN. The split is random, with the exception that molecules with less than eight heavy atoms were confined to the training subset. This dataset is available in the [supplementary material](#).

The 93 246 Psi4 computations were completed in a period of less than 24 h on Georgia Tech's Hive cluster.⁵⁹ The QC Fractal platform was used to evenly distribute jobs across 256 24-core nodes. Four jobs ran concurrently on every node. All results, including molecular orbitals, were deposited into the central Molecular Sciences Software Institute (MolSSI) QC Archive server.⁶⁰

B. MPNN implementation

There is much room for experimentation in the implementation of the CMPNN. We report a simple architecture that performs well for atomic multipoles. The hyperparameters present in this architecture were coarsely tuned, and an exhaustive optimization may yield further improvements in accuracy. The model described here is developed with the TensorFlow

2.3 library,⁶¹ and all associated code is publicly available for download.

The initial atomic hidden state vector \mathbf{h}_v^0 is formed by embedding the scalar atomic number Z_v into a vector of dimension 10. The edge feature vector \mathbf{e}_{vw} is formed from the scalar distance $|\mathbf{r}_{vw}|$ by projecting the distance onto a set of 43 Gaussian radial basis functions (RBFs). Each RBF has two associated parameters (μ, η) ,

$$RBF_i = e^{-\eta_i(|\mathbf{r}_{vw}| - \mu_i)^2}. \quad (22)$$

The distances μ_i are initialized uniformly between 0.8 and 5.0 Å, and all width parameters η_i are initialized to 100.0. The RBF parameters are free to optimize during training. RBFs are preferred to raw distances as they are an easier basis for learning non-linear functions. For distant pairs of atoms, the edge feature vector evaluates as the zero vector, making the atoms effectively non-interacting. Therefore, the number of actual edges in a molecular system scales asymptotically linearly with system size.

The message function M_t is defined as an outer product between the edge feature vector and the hidden state vector of the neighbor vertex; the current hidden state vector is not included in the message. The update function U_t , which determines a vertex's next hidden state, is a dense neural network composed of three hidden layers (256, 128, and 64 neurons) and an output layer of size 10. The dense neural network is applied to the concatenation of the vertex's current hidden state vector and message vector. Note that there separate weights are optimized for the monopole, dipole, and quadrupole tasks. Finally, the readout function is the sum of linear transformations of all hidden states, the weights of which are also made specific to the order of the multipole,

$$R = \sum_t \mathbf{w}^t \cdot \mathbf{h}_v^t + b^t. \quad (23)$$

It should be emphasized that each of these functions is a replaceable component of the model. The CMPNN reported in this work uses three message passing iterations, as we found no appreciable improvement beyond this value. The total system charge is conserved by uniformly adding or subtracting charge from all atoms. Similarly, traceless quadrupoles are enforced by uniformly adding or subtracting from the three diagonal quadrupole components.

There are additional choices associated with multitask aspect of the model. Since the CMPNN simultaneously models the atomic monopole, dipole, and quadrupole, the relative importance of each

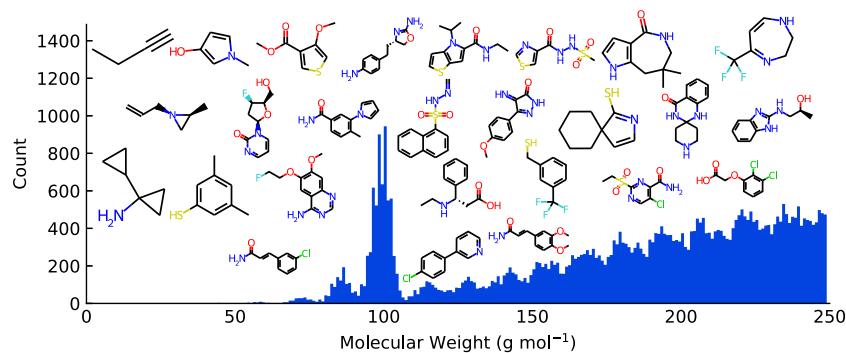


FIG. 2. A summary of the 46 623 molecule dataset. The distribution of molecular weights and a selection of representative structures are shown.

target's accuracy can be weighted by defining the loss function

$$\mathcal{L} = c^q \mathcal{L}^q + c^\mu \mathcal{L}^\mu + c^\Theta \mathcal{L}^\Theta, \quad (24)$$

where c^q , c^μ , and c^Θ are weights of the loss functions for the three tasks. We use the mean squared error for each individual loss function and set all loss coefficients to 1.0. The message, update, and readout functions for the dipole and quadrupole tasks are of the same form as the monopole task but with a different set of optimized weights. The quadrupole task is further split into predicting diagonal (Θ_{xx} , Θ_{yy} , Θ_{zz}) and off-diagonal (Θ_{xy} , Θ_{xz} , Θ_{yz}) components.

An 80%/10%/10% dataset split was used for training, validating, and testing on the dataset. The entire training procedure required ~5.5 days on a single P100 graphics processing unit (GPU). Once trained, a CMPNN model running on the same GPU can predict the atomic multipoles of a 40 atom molecule in 26 ms. Additional details about the training process can be found in the [supplementary material](#).

C. Online data augmentation

The directional CMPNN model proposed here is more flexible than the original rotation-invariant MPNN. Because the CMPNN only uses equivariant features and is not rigorously equivariant, a concern associated with this flexibility is that it may result in directional overfitting. A CMPNN trained on a specific molecule will correctly predict the dipole and quadrupole of the molecule in that same orientation. However, we cannot know that the model will make equally correct predictions if presented with the molecule rotated into a different orientation. This behavior is at odds with rotational equivariance; an ideal model would correctly predict dipoles and quadrupoles regardless of the orientation.

The concern of directional overfitting can be alleviated by a strategy referred to as "data augmentation," illustrated in Fig. 3. Data augmentation is the process of supplementing a training dataset with additional data derived by performing simple operations on the original training data. In the case of image models, this might be mirroring or rotating images.⁶² A useful form of data augmentation for the CMPNN is to randomly rotate the training molecules (and the corresponding atomic multipoles). This data augmentation could be performed offline, i.e., a few rotated copies of each molecule are created and appended to the training dataset. Instead, we perform online augmentation, which entails randomly rotating every

molecule each time the CMPNN is trained on it. This strategy is more memory efficient than offline data augmentation, and it ultimately exposes the CMPNN to more orientations of each molecule. For every batch of molecules of every epoch, we uniformly generate a random 3D rotation matrix and rotate all molecules in the batch. This minimizes overfitting and greatly improves generalization.

IV. RESULTS AND DISCUSSION

A. Multipole prediction

The accuracy of the atomic multipoles predicted by the CMPNN with respect to reference MBIS multipoles from PBE0/aug-cc-pV(T+d)Z densities is shown in Fig. 4. Over the entire test set, the model achieves a mean absolute error (MAE) of 0.003 (e) for monopoles, 0.002 ($e \cdot a_0$) for dipole vector components, and 0.003 ($e \cdot a_0^2$) for quadrupole tensor components. High accuracy on these test molecules, which were entirely excluded from the training procedure, verifies that the CMPNN has learned a general representation of atomic electrostatics for drug-like molecules. Not only are MAEs low but also the maximum errors are small and the distribution of errors appears to be unsystematic. The correlation between predicted and reference multipoles indicates that the higher order multipole components are slightly more difficult for the CMPNN to predict.

The saturation of the model with respect to the amount of training data used is shown in Fig. 5. The [supplementary material](#) contains additional element-specific learning curves. We see that the accuracy of all three multipole moments continues to improve with more data, suggesting that the CMPNN would benefit from an even larger and more diverse dataset. The CMPNN can be indirectly compared to the KRR model of Bereau *et al.*,²² which also predicted MBIS multipoles, albeit using a different dataset. The performance of the two models is comparable for small datasets, but the CMPNN errors quickly diminish with additional training data. The CMPNN is also relatively better at predicting the higher order multipoles, likely because of the use of rotationally equivariant and not invariant features.

B. Multipole orientation dependence

The CMPNN uses explicit Cartesian information to predict atomic dipoles and quadrupoles, which, because the model is not rigorously equivariant, may cause prediction accuracy to depend on the orientation of the molecule. The training data were augmented with random rotations in order to minimize this behavior. We test the effectiveness of this strategy in Fig. 6 by predicting the atomic dipole vector of the oxygen atom ($\bar{\mu}_O$) in *N*-methylacetamide (NMA) at 100 randomly generated orientations. According to the reference MBIS calculations, the oxygen dipole vector has a magnitude of ~0.085 $e \cdot a_0$ and is nearly aligned with the carbonyl bond, making an angle of about 4.6°. Across all 100 orientations, the CMPNN slightly but consistently overestimates the magnitude of this dipole vector, with predictions ranging from 0.087 $e \cdot a_0$ to 0.091 $e \cdot a_0$. The average predicted dipole angle (relative to the axis of the carbonyl bond) is similarly slightly but consistently underestimated, with predictions from 1.9° to 4.2°. All 100 predictions are reasonable when compared to the true MBIS values, given that this geometry does not appear

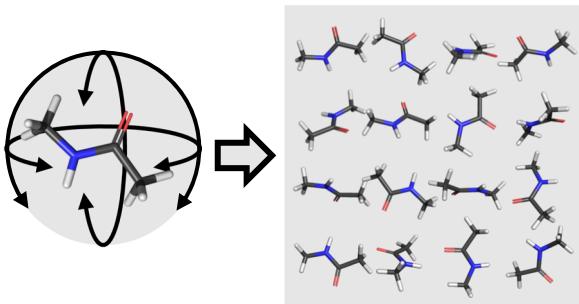


FIG. 3. Illustration of data augmentation by rigid rotation. By randomly rotating all training molecules every epoch, the CMPNN is forced to make accurate predictions across all molecular orientations.

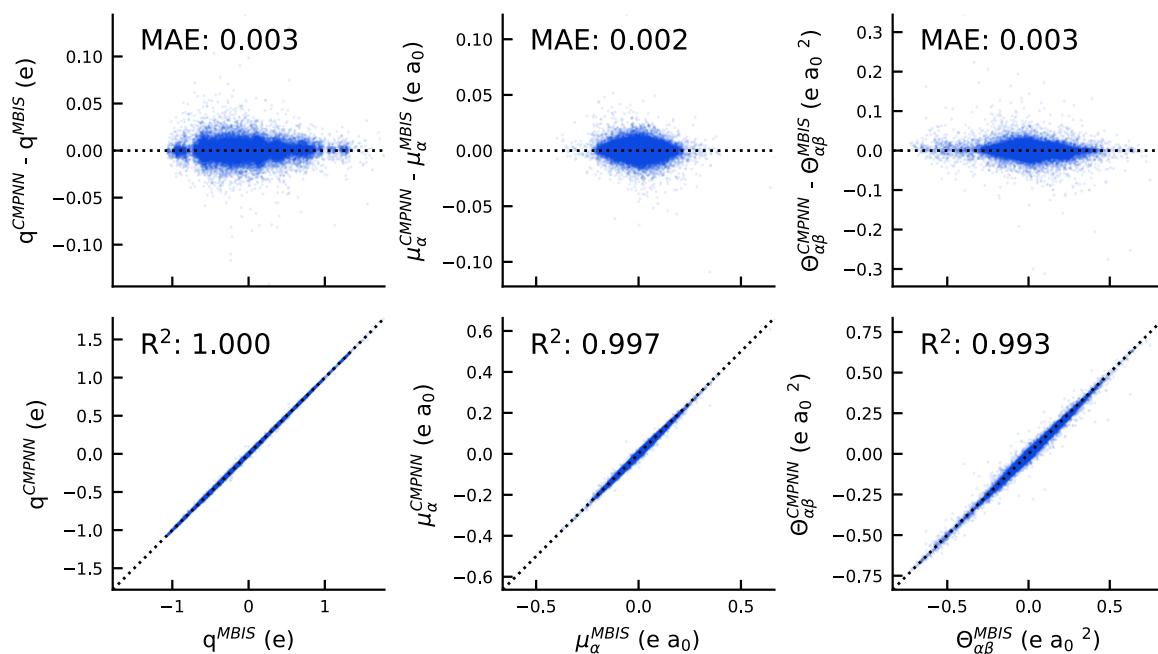


FIG. 4. Atomic multipoles predicted by the CMPNN are compared with reference MBIS multipoles over the 4663 molecule (116 204 atom) test set at the PBE0/aug-cc-pV(T+d)Z level of theory. The CMPNN prediction error (top) and CMPNN correlation with MBIS (bottom) is shown for atomic monopoles (left), dipole vector components (center), and Cartesian quadrupole tensor components (right).

in the training dataset. More importantly, there is little variability between predictions, suggesting that the representation of atomic multipoles learned by the CMPNN is approximately rotationally equivariant.

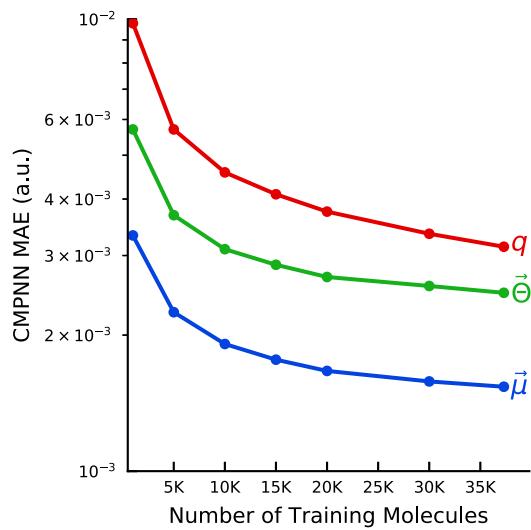


FIG. 5. Learning curve for CMPNN predictions of monopoles, dipole components, and quadrupole components. CMPNN models were trained on various fractions of the dataset, and MAEs were computed on the test set for each model.

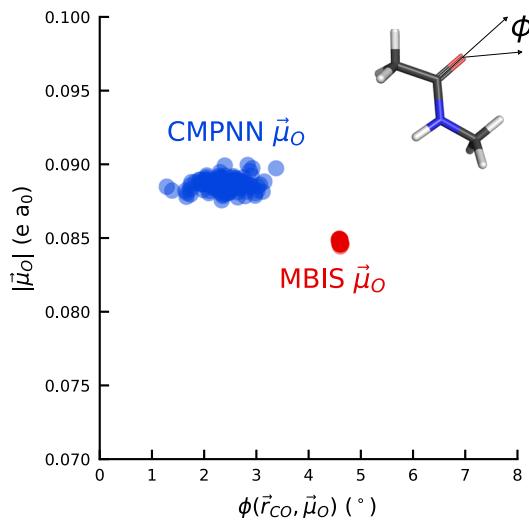


FIG. 6. A test of the orientation dependence of the CMPNN's predictions. The CMPNN is used to predict the dipole of the oxygen atom in NMA (pictured in the upper right corner) across 100 randomly generated reference frames. The predicted atomic dipole vector at each rotation is compared to the calculated MBIS atomic dipole at the PBE0/aug-cc-pV(T+d)Z level of theory. The dipole magnitude ($|\vec{\mu}_o|$) and angle relative to the molecule's carbonyl bond (ϕ) are used for this comparison, as both quantities should be independent of the orientation of the coordinate system.

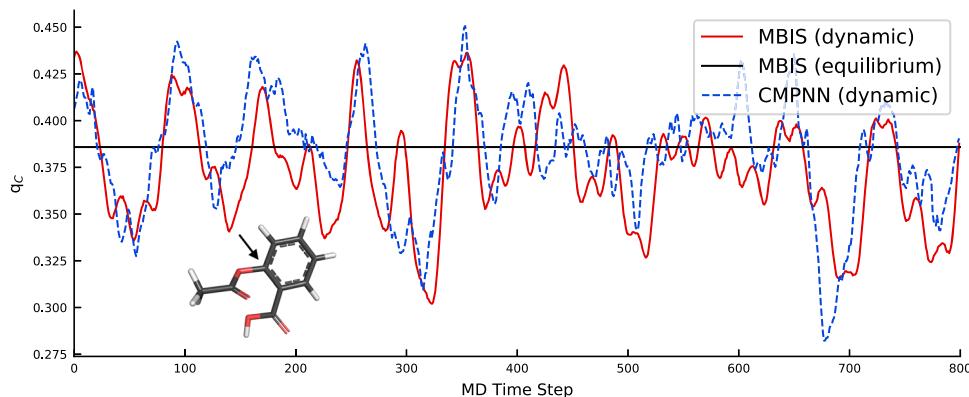


FIG. 7. A test of the CMPNN's ability to model different conformations of the same molecule. The atomic charge of the ester-adjacent aromatic carbon of the aspirin molecule (shown in the lower left corner) is calculated and predicted for each of the first 800 time steps of a MD trajectory.⁶³ For reference, the atomic charge from the optimized geometry is also shown.

C. Multipole conformational dependence

Using the CMPNN to model atomic properties allows one to capture conformationally dependent changes of each atom's local environment. The potential impact of this flexibility is demonstrated in Fig. 7, in which the atomic charge of a carbon on the aspirin molecule is modeled for a segment of a molecular dynamics (MD) trajectory. Although the partial charge on the carbon as computed with MBIS is +0.39 at the optimized aspirin geometry, the partial charge fluctuates between +0.29 and +0.44 throughout the first 800 time steps. This large range of values conveys the potential importance of conformation-dependent force field parameters in MD simulations. The CMPNN predictions approximately match the reference MBIS values throughout the trajectory and are, on average, a significant improvement over the static optimized-geometry charge. Note that this result is achieved without any aspirin conformations present in the training data. Adding a few non-equilibrium aspirin conformations to the dataset would likely improve the CMPNN predictions. The [supplementary material](#) contains a similar

demonstration that uses a small ensemble of three CMPNN models, providing an approximate measure of model uncertainty.

D. Electrostatic energy prediction

In Sec. IV A, we examined the accuracy of the multipoles predicted by the CMPNN. These predictions are only valuable to the extent that multipole errors are not magnified in quantities derived from the multipoles, namely, electrostatic interaction energies. We examine this propagation of errors in the multipole electrostatic interaction energy of the doubly hydrogen-bound formamide dimer along the dissociation coordinate from the HBC6 dataset.⁶⁴ The dimer geometry is optimized at each point in the dissociation coordinate, so this test requires conformationally dependent multipoles. Figure 8 shows that the MBIS and CMPNN multipoles, both computed and predicted separately for each monomer at the HF/jun-cc-pV(D+d)Z level of theory, yield similar electrostatic energies. Even at the closest dimer geometry, at which the carbons are separated

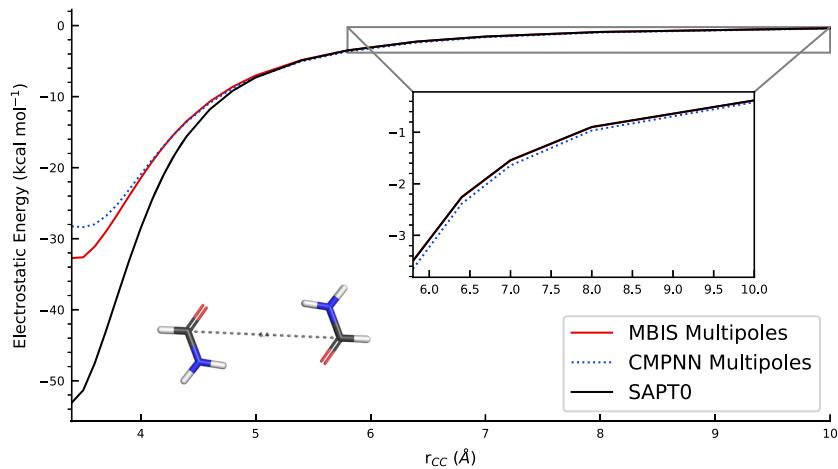


FIG. 8. Atomic multipoles (predicted by the CMPNN and computed with MBIS) are used to calculate the electrostatic interaction energy of the formamide dimer across the dissociation coordinate, which is defined by r_{CC} and has a minimum near 4.0 Å. Multipole electrostatics are compared with the significantly more accurate SAPTO electrostatic component. The CMPNN-predicted multipoles and reference MBIS multipoles yield similar electrostatic energies across all geometries. At close separations, the multipole electrostatics, which lack charge penetration, are underbound compared with SAPTO.

by 3.4 Å, the CMPNN multipole electrostatics differs by 4.5 kcal mol⁻¹ from the MBIS multipole electrostatics of 32.8, a 14% error. Both the absolute and relative errors improve along the dissociation coordinate as the small differences between the CMPNN and MBIS multipoles become less impactful. Beyond 3.8 Å, the CMPNN multipole electrostatics error relative to the MBIS multipole electrostatics is consistently within 1.0 kcal mol⁻¹ (6%), and by 5.8 Å, it decreases to 0.1 kcal mol⁻¹ (2%).

Regardless of the source of the multipoles, calculating electrostatics from the truncated multipole expansion is an approximation in and of itself, since both high-order multipole moments and charge penetration effects are not accounted for. As a reference, the exact electrostatic interaction energy of a dimer (at a given level of theory) can be computed with symmetry-adapted perturbation theory (SAPT) in the same jun-cc-pV(D+d)Z basis. Figure 8 also shows that the difference between the MBIS multipole electrostatics and CMPNN multipole electrostatics is quite small compared to the difference between either source of multipoles and exact SAPT electrostatics. Thus, the errors associated with the CMPNN multipoles are much smaller than errors arising from the use of the multipole expansion. If used in an *ab initio* force field, the short-range discrepancy between CMPNN multipole electrostatics and the reference SAPT electrostatics would be described by a separate charge penetration term.

V. CONCLUSIONS

The MPNN framework yields state of the art results in the prediction of many chemical properties. However, standard MPNN models are incapable of modeling directional atomic properties. Here, we introduce an extension of the MPNN framework to address this deficiency, and the newly developed CMPNN is used to predict Cartesian tensors, namely, atomic charges, dipoles, and quadrupoles. A multitask architecture is used to simultaneously predict all three atomic properties, and equivariance of the higher order atomic multipoles is approximately achieved by performing rotational data augmentation during training.

The CMPNN is accompanied by a newly developed and easily accessible dataset of 46 623 small, neutral, drug-like molecules and corresponding atom-centered multipoles calculated from both PBE0/aug-cc-pV(T+d)Z and HF/jun-cc-pV(D+d)Z densities. The fully trained CMPNN can predict atom-centered multipoles of unseen molecules with very high accuracy. We demonstrate that any small errors in the predicted multipoles lead to similarly small errors in multipole–multipole electrostatic interaction energy. In fact, the greatest source of error in multipole–multipole electrostatics arises from using the multipole approximation itself. Compared to reference quantum chemistry calculations, the CMPNN approach is orders of magnitude faster and scales asymptotically linearly with system size, meaning MD simulations of large systems could potentially use dynamically updated atomic multipoles. We expect this model to be of use in the development and parameterization of high-accuracy *ab initio* force fields.

SUPPLEMENTARY MATERIAL

The [supplementary material](#) contains additional information about the generation and chemical diversity of the dataset, details on

the training of the CMPNN model, and example code for accessing computed properties from the MolSSI server.

ACKNOWLEDGMENTS

The authors gratefully acknowledge financial support from Bristol Myers Squibb, the U.S. National Science Foundation (Grant No. CHE-1955940), and the U.S. Department of Energy (DOE) Office of Science, Office of Basic Energy Sciences, Computational Chemical Sciences (CCS) Research Program under work Proposal No. AL-18-380-057. We thank Dr. Ben Pritchard at MolSSI for guidance in using the QCArchive platform and depositing the dataset into the MolSSI QCArchive server. Computations were performed using the Georgia Tech Hive computer, funded by the U.S. National Science Foundation through Grant No. MRI-1828187 and maintained by the Partnership for an Advanced Computing Environment (PACE) at Georgia Tech.

DATA AVAILABILITY

The code to create a Cartesian MPNN model, the trained Cartesian MPNN models, and the coordinates and multipoles of the 46 623 molecule dataset used in this work are all publicly accessible on GitHub (see Ref. 65). These calculations were also deposited into the central MolSSI QCArchive server.⁶⁰

REFERENCES

- D. A. Case, I. Y. Ben-Shalom, S. R. Brozell, D. S. Cerutti, T. E. Cheatham III, V. W. D. Cruzeiro, T. A. Darden, R. E. Duke, D. Ghoreishi, M. K. Gilson, H. Gohlke, A. W. Goetz, D. Greene, R. Harris, N. Homeyer, S. Izadi, A. Kovalenko, T. Kurtzman, T. S. Lee, S. LeGrand, P. Li, C. Lin, J. Liu, T. Luchko, R. Luo, D. J. Mermelstein, K. M. Merz, Y. Miao, G. Monard, C. Nguyen, H. Nguyen, I. Omelyan, A. Onufriev, F. Pan, R. Qi, D. R. Roe, A. Roitberg, C. Sagui, S. Schott-Verdugo, J. Shen, C. L. Simmerling, J. Smith, R. Salomon-Ferrer, J. Swails, R. C. Walker, J. Wang, H. Wei, R. M. Wolf, X. Wu, L. Xiao, D. M. York, and P. A. Kollman, AMBER 2018, University of California, San Francisco, 2018.
- B. R. Brooks, C. L. Brooks, A. D. Mackerell, L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodosek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, and M. Karplus, *J. Comput. Chem.* **30**, 1545 (2009).
- E. Harder, W. Damm, J. Maple, C. Wu, M. Reboul, J. Y. Xiang, L. Wang, D. Lupyan, M. K. Dahlgren, J. L. Knight, J. W. Kaus, D. S. Cerutti, G. Krilov, W. L. Jorgensen, R. Abel, and R. A. Friesner, *J. Chem. Theory Comput.* **12**, 281 (2015).
- M. S. Gordon, M. A. Freitag, P. Bandyopadhyay, J. H. Jensen, V. Kairys, and W. J. Stevens, *J. Phys. Chem. A* **105**, 293 (2001).
- G. A. Cisneros, J.-P. Piquemal, and T. A. Darden, *J. Chem. Phys.* **123**, 044109 (2005).
- N. Gresh, P. Claverie, and A. Pullman, *Theor. Chim. Acta* **66**, 1 (1984).
- P. Xu, E. B. Guidez, C. Bertoni, and M. S. Gordon, *J. Chem. Phys.* **148**, 090901 (2018).
- P. K. Gurunathan, A. Acharya, D. Ghosh, D. Kosenkov, I. Kaliman, Y. Shao, A. I. Krylov, and L. V. Slipchenko, *J. Phys. Chem. B* **120**, 6562 (2016).
- U. Koch, P. L. A. Popelier, and A. J. Stone, *Chem. Phys. Lett.* **238**, 253 (1995).
- U. Koch and A. J. Stone, *J. Chem. Soc., Faraday Trans.* **92**, 1701 (1996).
- M. G. Darley, C. M. Handley, and P. L. A. Popelier, *J. Chem. Theory Comput.* **4**, 1435 (2008).
- R. Zubatyuk, J. S. Smith, J. Leszczynski, and O. Isayev, *Sci. Adv.* **5**, eaav6490 (2019).
- G. Csányi, T. Albaret, M. C. Payne, and A. De Vita, *Phys. Rev. Lett.* **93**, 175503 (2004).

- ¹⁴J. Behler and M. Parrinello, *Phys. Rev. Lett.* **98**, 146401 (2007).
- ¹⁵J. Behler, *J. Chem. Phys.* **134**, 074106 (2011).
- ¹⁶A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, *Phys. Rev. Lett.* **104**, 136403 (2010); [arXiv:0910.1019](https://arxiv.org/abs/0910.1019).
- ¹⁷K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, *Nat. Commun.* **8**, 13890 (2017).
- ¹⁸J. S. Smith, O. Isayev, and A. E. Roitberg, *Chem. Sci.* **8**, 3192 (2017).
- ¹⁹N. Lubbers, J. S. Smith, and K. Barros, *J. Chem. Phys.* **148**, 241715 (2018).
- ²⁰J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *International Conference on Machine Learning* (PMLR, 2017), pp. 1263–1272.
- ²¹J. S. Smith, B. Nebgen, N. Lubbers, O. Isayev, and A. E. Roitberg, *J. Chem. Phys.* **148**, 241733 (2018).
- ²²T. Bereau, R. A. DiStasio, A. Tkatchenko, and O. A. von Lilienfeld, *J. Chem. Phys.* **148**, 241706 (2018).
- ²³K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, *J. Chem. Phys.* **148**, 241722 (2018).
- ²⁴J. S. Smith, B. T. Nebgen, R. Zubatyuk, N. Lubbers, C. Devereux, K. Barros, S. Tretiak, O. Isayev, and A. E. Roitberg, *Nat. Commun.* **10**, 2903 (2019).
- ²⁵O. T. Unke and M. Meuwly, *J. Chem. Theory Comput.* **15**, 3678 (2019).
- ²⁶Z. L. Glick, D. P. Metcalf, A. Koutsoukas, S. A. Spronk, D. L. Cheney, and C. D. Sherrill, *J. Chem. Phys.* **153**, 044112 (2020).
- ²⁷J. Klicpera, J. Groß, and S. Günnemann, in *International Conference on Learning Representations* (ICLR), 2020.
- ²⁸A. Stone, *The Theory of Intermolecular Forces* (Oxford University Press, 2013).
- ²⁹J. W. Ponder, C. Wu, P. Ren, V. S. Pande, J. D. Chodera, M. J. Schnieders, I. Haque, D. L. Mobley, D. S. Lambrecht, R. A. DiStasio, M. Head-Gordon, G. N. I. Clark, M. E. Johnson, and T. Head-Gordon, *J. Phys. Chem. B* **114**, 2549 (2010).
- ³⁰T. Bereau, D. Andrienko, and O. A. von Lilienfeld, *J. Chem. Theory Comput.* **11**, 3225 (2015).
- ³¹A. Glielmo, P. Sollich, and A. De Vita, *Phys. Rev. B* **95**, 214302 (2017).
- ³²A. Grisafi, D. M. Wilkins, G. Csányi, and M. Ceriotti, *Phys. Rev. Lett.* **120**, 036002 (2018); [arXiv:1709.06757](https://arxiv.org/abs/1709.06757).
- ³³M. Veit, D. M. Wilkins, Y. Yang, R. A. DiStasio, and M. Ceriotti, *J. Chem. Phys.* **153**, 024113 (2020); [arXiv:2003.12437](https://arxiv.org/abs/2003.12437).
- ³⁴D. M. Wilkins, A. Grisafi, Y. Yang, K. U. Lao, R. A. DiStasio, and M. Ceriotti, *Proc. Natl. Acad. Sci. U. S. A.* **116**, 3401 (2019).
- ³⁵N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley, [arXiv:1802.08219](https://arxiv.org/abs/1802.08219) (2018).
- ³⁶B. Anderson, T. S. Hy, and R. Kondor, *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2019), Vol. 32.
- ³⁷B. K. Miller, M. Geiger, T. E. Smidt, and F. Noé, [arXiv:2008.08461](https://arxiv.org/abs/2008.08461) (2020).
- ³⁸S. Batzner, T. E. Smidt, L. Sun, J. P. Mailoa, M. Kornbluth, N. Molinari, and B. Kozinsky, [arXiv:2101.03164](https://arxiv.org/abs/2101.03164) (2021).
- ³⁹V. G. Satorras, E. Hoogeboom, and M. Welling, [arXiv:2102.09844](https://arxiv.org/abs/2102.09844) (2021).
- ⁴⁰K. T. Schütt, O. T. Unke, and M. Gastegger, [arXiv:2102.03150](https://arxiv.org/abs/2102.03150) (2021).
- ⁴¹A. J. Stone, *Chem. Phys. Lett.* **83**, 233 (1981).
- ⁴²A. J. Stone and M. Alderton, *Mol. Phys.* **56**, 1047 (1985).
- ⁴³T. Verstraelen, S. Vandenbrande, F. Heidar-Zadeh, L. Vanduyfhuys, V. Van Speybroeck, M. Waroquier, and P. W. Ayers, *J. Chem. Theory Comput.* **12**, 3894 (2016).
- ⁴⁴F. L. Hirshfeld, *Theor. Chim. Acta* **44**, 129 (1977).
- ⁴⁵F. Heidar-Zadeh, P. W. Ayers, T. Verstraelen, I. Vinogradov, E. Vöhringer-Martinez, and P. Bultinck, *J. Phys. Chem. A* **122**, 4219 (2017).
- ⁴⁶Z. Qiao, M. Welborn, A. Anandkumar, F. R. Manby, and T. F. Miller, [arXiv:2007.08026](https://arxiv.org/abs/2007.08026) (2020).
- ⁴⁷R. Caruana, *Mach. Learn.* **28**, 41 (1997).
- ⁴⁸C. Zhang, C. Lu, Z. Jing, C. Wu, J.-P. Piquemal, J. W. Ponder, and P. Ren, *J. Chem. Theory Comput.* **14**, 2084 (2018).
- ⁴⁹M. J. Van Vleet, A. J. Misquitta, and J. R. Schmidt, *J. Chem. Theory Comput.* **14**, 739 (2018).
- ⁵⁰D. Mendez, A. Gaulton, A. P. Bento, J. Chambers, M. De Veij, E. Félix, M. P. Magariños, J. F. Mosquera, P. Mutowo, M. Nowotka, M. Gordillo-Marañón, F. Hunter, L. Junco, G. Mugumbate, M. Rodriguez-Lopez, F. Atkinson, N. Bosc, C. J. Radoux, A. Segura-Cabrera, A. Hersey, and A. R. Leach, *Nucleic Acids Res.* **47**, D930 (2019).
- ⁵¹C. A. Lipinski, F. Lombardo, B. W. Dominy, and P. J. Feeney, *Adv. Drug Delivery Rev.* **46**, 3 (2001).
- ⁵²C. A. Lipinski, *Drug Discovery Today: Technol.* **1**, 337 (2004).
- ⁵³Schrödinger Release 2020-2: MacroModel, Schrödinger, LLC, New York, NY, 2020.
- ⁵⁴L. C. Blum and J.-L. Reymond, *J. Am. Chem. Soc.* **131**, 8732 (2009).
- ⁵⁵M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. v. Lilienfeld, *Phys. Rev. Lett.* **108**, 058301 (2012).
- ⁵⁶D. G. A. Smith, L. A. Burns, A. C. Simonett, R. M. Parrish, M. C. Schieber, R. Galvelis, P. Kraus, H. Kruse, R. Di Remigio, A. Alenaizan, A. M. James, S. Lehtola, J. P. Misiewicz, M. Scheurer, R. A. Shaw, J. B. Schirber, Y. Xie, Z. L. Glick, D. A. Sirianni, J. S. O’Brien, J. M. Waldrop, A. Kumar, E. G. Hohenstein, B. P. Pritchard, B. R. Brooks, H. F. Schaefer, A. Y. Sokolov, K. Patkowski, A. E. DePrince, U. Bozkaya, R. A. King, F. A. Evangelista, J. M. Turney, T. D. Crawford, and C. D. Sherrill, *J. Chem. Phys.* **152**, 184108 (2020).
- ⁵⁷P. Bleiziffer, K. Schaller, and S. Riniker, *J. Chem. Inf. Model.* **58**, 579 (2018).
- ⁵⁸T. Verstraelen, P. Tecmer, F. Heidar-Zadeh, E. C. González-Espinoza, M. Chan, T. D. Kim, K. Boguslawski, S. Fias, S. Vandenbrande, D. Berrocal, and W. Paul, Ayers HORTON 2.1.1, <http://theochem.github.com/horton/>, 2017.
- ⁵⁹PACE, Partnership for an Advanced Computing Environment (PACE), 2017.
- ⁶⁰D. G. A. Smith, D. Altarawy, L. A. Burns, M. Welborn, L. N. Naden, L. Ward, S. Ellis, B. P. Pritchard, and T. D. Crawford, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **11**, e1491 (2021).
- ⁶¹M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org.
- ⁶²C. Shorten and T. M. Khoshgoftaar, *J. Big Data* **6**, 60 (2019).
- ⁶³S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, *Sci. Adv.* **3**, e1603015 (2017).
- ⁶⁴K. S. Thanthiriwatte, E. G. Hohenstein, L. A. Burns, and C. D. Sherrill, *J. Chem. Theory Comput.* **7**, 88 (2010).
- ⁶⁵Z. L. Glick, Cartesian MPNN, <https://github.com/zachglick/directional-mpnn>, 2021.