# Machine Learning Transferable Physics-Based Force Fields using Graph Convolutional Neural Networks

by

William H. Harris

B.S. Chemical Engineering, B.S. Physics
North Carolina State University, 2016

SUBMITTED TO THE DEPARTMENT OF MATERIALS SCIENCE AND ENGINEERING
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN MATERIALS SCIENCE AND ENGINEERING
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

SEPTEMBER 2020



Signature of Author: _____

Department of Materials Science and Engineering
August 6, 2020

Certified by: _____

Rafael Gomez-Bombarelli
Assistant Professor of Materials Science and Engineering
Thesis Supervisor

Accepted by: _____

Frances Ross
Professor of Materials Science and Engineering
Chair, Departmental Committee on Graduate Studies

Machine Learning Transferable Physics-Based Force Fields using
Graph Convolutional Neural Networks

by
William H. Harris

Submitted to the Department of Materials Science and Engineering
on August 6, 2020 in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Materials Science and Engineering

ABSTRACT

Molecular dynamics and Monte Carlo methods allow the properties of a system to be determined from its potential energy surface (PES). In the domain of crystalline materials, the PES is needed for electronic structure calculations, critical for modeling semiconductors, optical, and energy-storage materials. While first principles techniques can be used to obtain the PES to high accuracy, their computational complexity limits applications to small systems and short timescales. In practice, the PES must be approximated using a computationally cheaper functional form. Classical force field (CFF) approaches simply define the PES as a sum over independent energy contributions. Commonly included terms include bonded (pair, angle, dihedral, etc.) and non-bonded (van der Waals, Coulomb, etc.) interactions, while more recent CFFs model polarizability, reactivity, and other higher-order interactions. Simple, physically-justified functional forms are often implemented for each energy type, but this choice – and the choice of which energy terms to include in the first place – is arbitrary and often hand-tuned on a per-system basis, severely limiting PES transferability. This flexibility has complicated the quest for a universal CFF. The simplest usable CFFs are tailored to specific classes of molecules and have few parameters, so that they can be optimally parameterized using a small amount of data; however, they suffer low transferability. Highly-parameterized neural network potentials can yield predictions that are extremely accurate for the entire training set; however, they suffer over-fitting and cannot interpolate. We develop a tool, called AuTopology, to explore the trade-offs between complexity and generalizability in fitting CFFs; focus on simple, computationally fast functions that enforce physics-based regularization and transferability; use message-passing neural networks to featurized molecular graphs and interpolate CFF parameters across chemical space; and utilize high performance computing resources to improve the efficiency of model training and usage. A universal, fast CFF would open the door to high-throughput virtual materials screening in the pursuit of novel materials with tailored properties.

Thesis Supervisor: Rafael Gomez-Bombarelli
Title: Assistant Professor of Materials Science and Engineering

# Table of Contents

# Introduction

Molecular dynamics (MD) and Monte Carlo (MC) methods are established simulation techniques that remain the computational workhorses of the materials science community, allowing thermodynamic and kinetic properties of a system to be determined from its potential energy surface (PES). Within the domain of crystalline materials, moreover, the PES provides the necessary input for electronic structure calculations, which are critical for modeling semiconductors, optical and energy-storage materials, and other functional materials whose target properties depend strongly on electronic interactions. In principle, these methods yield results that are arbitrarily exact up to the accuracy of the provided PES. However, while first principles-based quantum chemistry techniques such as coupled cluster methods can be used to obtain the PES to high accuracy, their computational complexity limits applications to relatively small systems and short timescales. Therefore, for practical systems of even moderate size, the PES must be approximated as a function of the system configuration using a computationally cheaper functional form for the Hamiltonian.

Classical force field (CFF) approaches – by far the most common for large systems because of their computational efficiency – simply define the PES as a sum over independent energy contributions, each of which aims to capture some aspect of the underlying physics. Commonly included terms include those for describing bonded (pair, angle, dihedral, etc.) and non-bonded (van der Waals, Coulomb, etc.) interactions, while more recent CFFs have added terms to model polarizability, reactivity, and other higher-order interactions. Simple, physically-justified functional forms are often implemented for each energy type, such as 1D harmonics for bond energies, but this choice – and the choice of which energy terms to include in the first place – is ultimately arbitrary and often hand-tuned on a per-system or per-application basis, thus severely limiting PES transferability.

This large flexibility in implementing CFFs – namely, the flexibility to adjust the functional form/complexity of each energy contribution and to decide which of those terms, and therefore what physics, to include in the system – has complicated the quest for a truly universal classical force field. At one extreme, the simplest usable CFFs are tailored to specific classes of molecules and have few parameters, so that they can be optimally parameterized using a small amount of training data; however, they suffer from low generalizability (a CFF trained on hydrocarbons will not perform well for alcohols). On the other hand, recently published highly-parameterized neural network potentials have such a large representational capacity that they can yield energy predictions that are extremely accurate for the entire training set, no matter how chemically diverse; however, these models suffer from over-fitting, so that they cannot reliably interpolate to unseen points in chemical configuration space (CCS). In this project, we develop a tool, called AuTopology, to (1) systematically explore the trade-offs between functional complexity and generalizability in fitting CFFs; (2) focus on simple, computationally fast functional forms that enforce physics-based regularization and transferability; (3) use message-passing neural networks to featurized molecular graphs and interpolate CFF parameters across chemical space; and (4),

whenever possible, utilize high performance computing resources to improve the efficiency of model training and usage.

Once its functional form is chosen, each energy term is parameterized based on the identity of the interacting atoms. For example, a pair interaction may be modeled as a harmonic bond, where the force constant $k$ and equilibrium bond length $r_0$ are specified in a look-up table and are different for a C-O single bond, a C-C single bond, and a C=C double bond, just to name a few. Importantly, the accuracy of CFF methods can be improved by parameterizing interactions in a way that distinguishes different chemical environments: for instance, chemical intuition suggests that the C-C bond in ethane is chemically distinct from the C-C bond in ethanol due to the polarization induced by the adjacent oxygen atom, and should therefore have different values for bond stiffness and equilibrium length. The number of chemically distinct interactions grows as the radius $R$ of the chemical neighborhood – itself a tunable hyperparameter of the model – grows. For example, $R \geq 2$ is enough to distinguish all C-H bonds in ethanol from those in ethane. The correspondingly large number of adjustable parameters in the CFF model are generally fit using a mixture of ab initio and experimental techniques. For example, bond and angle parameters can often be determined from experimental vibrational spectra, Lennard-Jones radii from crystal structure packing densities, and partial charges for Coulomb interactions from electronic structure calculations on molecular fragments. Attempts to improve the transferability of these models have led to the conclusion that PESs produced this way can be reliably applied only when (a) the parameters are, whenever possible, fit indirectly via the bulk properties of interest, and (b) are fit to specific material classes, such as proteins or semiconductors – but not both.

To date, this transferability problem has not been overcome or systematically studied despite the significant benefits that such a PES model would provide. Therefore, with the goal of addressing an important and outstanding need of the computational materials science community while adhering to the overarching project objectives listed previously, we develop a novel CFF model that seeks to meet the following criteria: it (1) is transferable, so that once the model is trained, its applicability extends to multiple materials classes; (2) can be fit to energy or forces without relying on measurements or predictions of bulk properties, thus reducing the burden of curating large training datasets; (3) provides a continuous and readily interpretable latent space through which chemical configurations (e.g., a C-C-O angle interaction within an acetyl group) are mapped to CFF parameters (e.g., bond stiffness and equilibrium angle), so that interpolation between training points is not only justified, but also provides insight into the underlying physics, as seen in the case of the C-C bond dipole moment mentioned above; (4) achieves higher accuracy relative to the quantum chemistry benchmark than current CFF parameterization methods, thus justifying the research effort in the first place; (5) is built from the start for continuous training and refinement of model parameters as new and larger datasets become available, so that the construction of the latent space should always be seen as a work in progress (filling blind spots, expanding frontiers, refining regions that are complex or of particular interest for a given application); (6) is provided, in its final form to end users, as an open-source tool, called AuTopology, that can be integrated seamlessly into existing computational workflows or molecular simulation packages, and (7)

retains the computational efficiency of existing CFF techniques, thus ensuring its continued practical applicability to the research community.

The training framework outlined above, as well as the criteria listed for the final model, suggest a machine learning (ML)-based approach for predicting classical force field parameters of interactions within arbitrary chemical environments. Motivated by recent advances in graph-based ML, we propose a graph convolutional neural network (GCNN) functional form for the CFF parameter predictor. Recent theoretical results show that distinguishing subgraphs (which is equivalent to the problem of distinguishing chemical neighborhoods) can be achieved using a message-passing neural network (MPNN), a specific type of GCNN, that learns an injective mapping from molecular subgraphs to their latent space embeddings that is at least as powerful as the Weisfeiler-Lehman graph isomorphism test. After $R$ convolutions, each node is labeled by a feature vector (of tunable length) that contains information about itself and its radius-$R$ neighborhood. These vectors serve as the inputs to feed-forward neural network(s) that predict the CFF parameters; given a three-dimensional molecular configuration, energy and forces can then be calculated and trained via backpropagation and gradient descent.

This project is tailored specifically to hook into high-throughput materials discovery pipelines, both during data generation, training, and final use of the trained model. As such, computational efficiency and memory requirements will be carefully considered at each stage so that high performance computing resources can be fully utilized whenever available. Training will be performed in parallel on GPUs and CPUs utilizing mixed precision where appropriate; numerical techniques that store interactions as neighbor lists instead of adjacency matrices will allow memory to scale linearly instead of quadratically with the system size, thus fitting more chemistry onto one GPU; the pipeline for obtaining DFT energies for training data will be deployable in a massively parallel setting, thus making it ripe for intensive data production campaigns.

# Background

## Graph Convolutional Neural Networks

ML methods have demonstrated great promise in leveraging the existing wealth of experimental and computational data in materials science to accelerate the development of novel materials with tailored properties. Artificial neural networks (ANNs) are trained to learn functions $F$ that map points $X \in \chi$ in the CCS to target properties; for example, $F$ could map a molecular conformation to its potential energy, an amino acid chain to its functional class, a protein-ligand pair to its binding energy, or a set of atomic configurations to its ensemble-averaged free energy. In all cases, the raw configuration, whether as atomic positions or a coarser-grained spatial representation, is transformed into a latent space embedding, $\mathrm{ENC}(X) = Z$, upon which the property predictor is

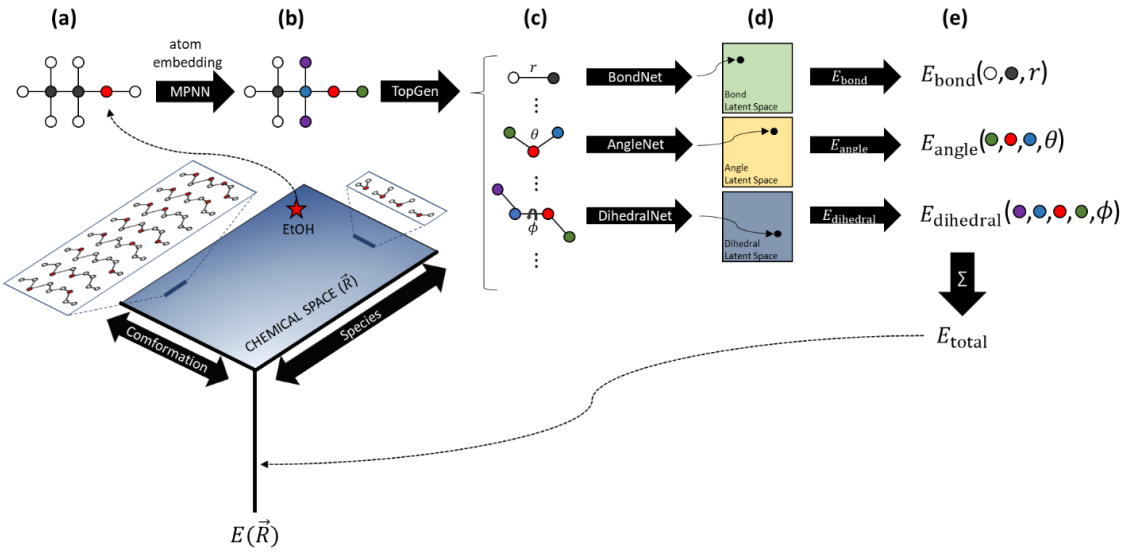


**Figure 1.** Calculation of molecular energies using a classical force field-based sum over bonded clusters

**(a)** Molecular graph $\mathcal{G}$ with physics-based node and edge features

**(b)** Node features updated by graph convolutions of MPNN that incorporate $R = 2$ neighborhood information

**(c)** Enumeration of all instances of each topology type $\tau \in$ {bond, angle, dihedral, …}

**(d)** For each $\tau$, an NN maps MPNN-featurized topology instances $T_i^{(\tau)} \in T^{(\tau)}$ to a latent space $\chi^{(\tau)}$

**(e)** The energy contribution of $T_i^{(\tau)}$ is determined using a physics-based function $E_\tau$ parameterized by $\chi^{(\tau)}$

trained. In order to accelerate training while improving the capacity of high-throughput materials discovery and virtual screening platforms, ENC should capture only those features of the input data which are relevant to the prediction task at hand; at the same time, it should be flexible enough that transferability to similar problems can be exploited whenever possible. One approach is to utilize domain-specific knowledge or heuristics to hand-craft the encoding functions. This approach has recently been used to encode protein pockets for enzyme classification prediction based on biochemical intuition: for instance, after voxelizing a three-dimensional protein structure, each voxel can be featurized with a value indicating intersection with the protein backbone [1] or with a feature vector indicating proximity to C, H, O, and N atoms [2]. Other projects included channels for hydrophobicity or descriptors capturing solvation effects [3] in order to encode task-relevant protein features.

However, in contrast to methods in which encoding is treated as a pre-processing step, the developing consensus is that the encoder itself should be learned from the data as part of the ML training pipeline. Such approaches not only show superior performance because they learn embeddings that are optimized for a specific prediction task [4], but *crucially* they also lend themselves to the problem of inverse design – an important pillar of modern materials science.

Using an autoencoder framework [5, 6], a decoder is jointly trained alongside $F$ to minimize (relative to some user-defined metric) the distance between $X$ and its reconstruction, $X' = \mathrm{DEC}(\mathrm{ENC}(X))$. Not all auto-encoders attempt to reconstruct the input; instead, the goal is to make the latent space more interpretable and easier for subsequent ANNs to learn from, such as ensuring that $\mathrm{dist}(X_1, X_2) \cong \mathrm{dist}(X_1', X_2')$ [7]. The use of variational autoencoders and stacked autoencoders has pushed various prediction tasks to their state-of-the-art [8, 9] and has opened a new avenue for approaching the property-to-structure engineering challenge.

The best performance on a range of molecular property prediction tasks has been achieved using graph convolutional network (GCN) encoders [7, 11-15]. These methods treat molecules as graphs, where nodes represent atoms, edges represent bonds, and each graph, node, and edge may possess a feature vector containing information such as atomic number, valence, bond type, and aromaticity. Like convolutional neural networks used in image classification, GCNs iteratively updates node feature vectors using a message-passing scheme that aggregates information from neighboring nodes and edges and combines them to produce an updated feature vector. After $k$ iterations or convolutions, each node possesses information spanning its radius-$k$ neighborhood. A global pooling operating produces a graph embedding, which in this case is interpreted as a molecular fingerprint, $f$.

While this approach has achieved state-of-the-art results in predicting molecular properties from their trained fingerprints – highlighting the benefits of representation learning – as yet there is no clear consensus on (1) how to merge the successes of the two previous paragraphs, (2) what kinds of materials systems can be effectively modeled as graphs and in what circumstances, and (3) how those graphs should be feauturized, especially when three-dimensional structure is important, as in protein-ligand binding. Furthermore, can chemical reactions be treated with a similar framework, where the trained latent space encodes a reduced set of reaction coordinates which, in combination with an ANN predicting potential energy, can be used to identify potentially new reaction pathways? How far can this idea of learned representations be pushed?

Indeed, it was only very recently that any work was published treating the theoretical limits of GCNs, such as their ability to distinguish different molecules (the graph isomorphism problem) [12]. This brings into focus the need for a firm theoretical foundation for the problem of inverse-design of molecular graphs. Moreover, it is often the case that certain input features to GCN methods can be predicted from other input features – in other words, there is redundant information in the training data [4]. Does this apply to bond lengths in molecular graphs? Can full 3D structure be inferred from a 2D representation of a molecule? Can multiple length scales be addressed in the same training pipeline? How can graph-based embeddings of multiscale materials systems be leveraged in an encoder-decoder framework to predict novel materials with targeted properties?

The inverse design challenge is two-fold: quantum mechanical calculations are slow, taking hundreds of CPU-hours for systems of even moderate size, and the CCS is extremely large. ANNs can be used – as part of an automated and high-throughout materials discovery pipeline – to train

from existing structure-property data and to extrapolate *in silico* to new regions of the CCS without costly and time-consuming experimental trial-and-error. However, identifying the encodings that will train a highly interpretable and practically useful latent space while maintain a representation that is simple enough to decode to a molecular graph is an open problem. If solved, this would provide a massive step forward in the quest to design novel materials with properties that have yet to be accessed experimentally. We hope that our AuTopology tool, using graph convolution-based encodings of molecular structures trained on the output of high-throughput electronic structure calculations, can be used to push the development of novel materials for energy and other applications. Although not done as part of this project, the goal is for others to leverage this tool to screen and suggest promising candidate materials that can be further studied by experimentalists, whose subsequent characterization can provide further data to the ML algorithms in an iteratively refined theory-to-experiment optimization loop.

## Partitioning Molecular Energies

In general, the energy of an arbitrary configuration of $N$ atoms in a multi-component system can be formally expanded as a sum over multi-body or $k$-body interactions [16], where $k = 1,2,\dots,N$.

$$E(\vec{r}_1,\vec{r}_2,\dots,\vec{r}_N) = \sum_i E_i^{(1)} + \frac{1}{2}\sum_i\sum_j E_{ij}^{(2)} + \frac{1}{3!}\sum_i\sum_j\sum_k E_{ijk}^{(3)} + \cdots = \sum_i E_i \quad (1)$$

$$E_i = E_i^{(1)} + \frac{1}{2}\sum_j E_{ij}^{(2)} + \frac{1}{3!}\sum_j\sum_k E_{ijk}^{(3)} + \cdots \quad (2)$$

The sums are unrestricted and indices run over all atoms. The final equality of Eq. 1 partitions $E$ into $N$ atom-wise contributions by defining an atomic energy, $E_i$, depending on the full chemical environment of each atom. In this case, we need only focus our attention on calculating $E_i$ via Eq. 2. Note, however, that if $E$ is approximated using less general functional forms than those given above, a separation into atomic contributions may not be possible. This provides an important classification of FFs into those that define an $E_i$ term and those that do not, a useful distinction in the context of this project.

The convergence of the sum in Eq. 2 is usually slow, especially for highly-correlated systems like bulk metals, and the number of interaction terms to compute quickly becomes prohibitive for systems of practical interest [17]. Restricting the sum by enforcing a cutoff distance reduces computational cost but risks neglecting important long-range energy contributions, such as electrostatic interactions in an ionic liquid or the many-body effects that characterize conjugated systems. Thus, developing a computationally tractable force field of this form requires choosing (1) the $k$-body interactions that are most important and (2) their functional forms. As explained in more detail below, the deficiencies of common FFs can be traced to overly restrictive decisions made in both categories.

The fundamental problem with Eq. 2 is that there is no physics-based justification for partitioning molecular energies into such $k$-body interactions. Indeed, because the form of the exchange-

correlation functional is unknown, no partitioning scheme is available for rigorously decomposing the energy of a quantum system into separate terms [18]. Fortunately, the situation is less dire in practice; many techniques, such as Energy Decomposition Analysis (EDA) [19] and the Interacting Quantum Atom (IQA) method [20], have been developed for partitioning molecular energies as approximated using various levels of quantum chemical theory. They take advantage of the simplifying assumptions underlying Hartree-Fock, Coupled Cluster, and other *ab initio* methods to rewrite Eq. 1 as an expansion over well-defined energy terms. Unlike black-box methods for calculating energy, this energy partitioning not only offers insight into the underlying physics but provides a bridge between quantum mechanics and the heuristic, high-level chemical intuition that must be reproduced by an interpretable energy prediction tool.

As a representative example, the IQA method, which partitions Hartree-Fock-calculated energies, starts by spatially dividing a molecule into non-overlapping regions called topological atoms (which are almost always centered on nuclei and therefore correspond to the usual notion of atoms) by taking the Laplacian of its electron charge distribution and identifying the basins of attraction of its gradient field [20]. Each topological atom is assigned an energy of the form

$$E_i = E_i^{\text{intra}} + \frac{1}{2}\sum_{j \neq i} E_{ij}^{\text{inter}} \tag{3}$$

$$E_{ij}^{\text{inter}} = E_{ij}^{\text{Coulomb}} + E_{ij}^{\text{XC}} = E_{ij}^{\text{nn}} + E_{ij}^{\text{en}} + E_{ij}^{\text{ee}} + E_{ij}^{\text{XC}} \tag{4}$$

The atomic self-energy depends only on the charge distribution within a basin and includes electron kinetic energy, self-repulsion, and nuclear attraction. The interatomic term is defined between all pairs of basins and contains pairwise Coulomb interactions between nuclei and electron charge distributions, in addition to the associated exchange-correlation energy [20]. The exact details are less important for the current discussion; what matter is that IQA and similar methods have been used to quantitatively investigate the relative importance of these various energy contributions in different chemical systems. Taken together, they begin to paint a clearer picture of the factors that limit the transferability and interpretability of many widely-used CFFs. However, before diving into these issues, we review the major force fields in the following section.

## Limitations of Classical Force Fields

Based on chemical intuition and relative historical success, modern FFs share a functional form which can be understood as extensions of the original AMBER force field [27]. This model separates total energy into bonded (bond, angle, torsional) and non-bonded (long-range electrostatic) interactions.

$$E = \sum_{\text{bonds}} k_r(r - r_0)^2 + \sum_{\text{angles}} k_\theta(\theta - \theta_0)^2 + \sum_{\text{torsions}} V_n[1 + \cos(n\varphi - \gamma)] + \sum_{\text{nb}} \left( \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} + \frac{q_i q_j}{\varepsilon r_{ij}} \right) \tag{5}$$

AMBER considers only dihedral torsions while CHARMM, otherwise identical in form, also considers improper torsions; this is primarily to enforce planar geometry around $sp^2$ centers [28].

The harmonic forms of bonded interactions are primarily motivated as first-order approximations to near-equilibrium geometries, while the non-bonded terms approximate Coulomb and dispersion interactions between fixed and induced point charges centered on nuclei. Within this framework, the general multi-body expansion in Eq. 1 is limited to 2-, 3-, and 4-body terms, with 2-body interactions further divided into bonded (short-range) and non-bonded (long-range) pairs. As a further simplification, all 3- and 4-body terms that are not represented in the molecular graph are neglected.

It is important to remember that although these *ad hoc* decisions significantly simplify the underlying physics, FFs can be often be parameterized to an adequate level of accuracy for specific applications. For example, the latest implementation of CHARMM has separate (non-transferable) parameters sets for proteins, nucleic acids, lipids, carbohydrates, and general organic molecules [32]. The force constants, torsion parameters, Lennard-Jones parameters, and partial charges are fit with a combination of *ab initio* calculations and Monte Carlo simulations of bulk thermodynamic properties. The OPLS force field is similar to CHARMM but was parameterized to provide better agreement with experimental liquid-phase properties and allows off-atom partial charges for better modelling of electrostatic interactions [29]. The AMBER/CHARMM type FFs (class I force fields) have been improved upon by using anharmonic bonded interactions (thus allowing greater bond extensions) [33] and by incorporating cross-terms that, for example, depend on more than one bond distance (class II force fields) [16].

We can utilize the notion of energy partitioning mentioned in the previous section to probe how adequately the functional forms of class I and class II force fields can be expected to fit molecular energies. By analyzing the IQA-partitioned energies of ethane, hydrogen peroxide, acetaldehyde, and five other simple organic molecules as a function of torsion angle, it has been demonstrated that the torsion terms in class I force fields have little chance of being transferable between chemical systems [30]. This is due to the complex interplay of different energetic contributions (kinetic, electrostatic, exchange-correlation, etc.) that they are attempting to model with only a few adjustable parameters. Fortunately, the same line of research has found that 1-3 (angle) and 1-4 (torsional) energy terms can be more accurately modeled using a multipole expansion of the electrostatic interaction, thus treating them on equal footing with long-range non-bonded interactions [31].

There are a wealth of further examples in the literature highlighting the futility of improving existing force fields without making changes to their underlying functional forms. For example, while the limitations of using point charges instead of multipole expansions to model electrostatic interactions has been well established [22], they continue to be used widely in the simulation community [24, 25].

Most modern FFs treat electrostatic interactions additively by assigning fixed full or partial charges to atoms and ions; this approach is inherently limiting because it neglects environment-induced polarization [34]. Although parameter sets fit to bulk properties may effectively absorb this error into existing force terms, their applicability breaks down for heterogeneous systems like charge-

surface interactions [35], ion solvation [36, 37], and protein folding in different dielectric environments [46]. Polarizability was included in the latest class of FFs, such as AMOEBA [44], using three main approaches: induced dipoles, classical Drude oscillators, and fluctuating charges [34]. Although more computationally demanding, explicit inclusion of polarization has improved agreement with experiment in studies of solvation energies [38, 40, 42], biomacromolecules in aqueous solution [39, 31], and (to a lesser extent) bulk water [43], and it is widely agreed that explicit inclusion of polarization effects will be crucial to the development of improved and transferable force fields [45, 47].

## Machine Learning Classical Force Fields

The previous sections have demonstrated the importance of an atom's extended chemical environment (beyond simply its bonded neighbors) on the parameterization of interaction terms in classical force fields. A transferable FF parameterization would therefore need a way to quantify an atom's environment that scales well with the total number of atoms (so that computational efficiency is maintained) while only keeping as much information as necessary. The latter consideration – of avoiding over-parameterization – is an important regularization method for fitting energies because it reduces the risk of under-determining the model given a fixed amount of training data.

Neural network potentials [48, 49, 50] have used continuous spatial kernels and atom-centered symmetry functions to obtain numerical fingerprints of an atom's chemical environment. These fingerprints, or embeddings, are simply vectors of real numbers that are passed through feed-forward neural networks and trained to predict $E_i$, as defined in Eq. 2. While NN potentials have demonstrated great success in fitting to *ab initio* energies, their functional forms have such a high representational capacity (that is, they are so flexible) that (1) they are uninterpretable, so that it is difficult to use them to gain insight into the underlying physics, and (2) they may fail spectacularly when interpolating or extrapolating to chemistry not seen in the training set.

This represents the opposite extreme from CFFs, whose failure to adequately fit energies is due to a lack of representational capacity and their use of insufficiently flexible functional forms. This project seeks to find a balance between these two extremes: interpretable, physics-based functional forms that introduce only as much functional complexity as is required to represent the underlying physics.

Graph convolutional neural networks (GCNNs) provide a convenient framework for learning embeddings of an atom's extended neighborhood. Intuitively, chemical bonds can be interpreted as "information paths" along a molecular graph, a notion that is well-supported by their utility in chemistry education. Therefore, defining an atom's radius-$R$ neighborhood as those atoms within $R$ bonds on the molecular graph, rather than those bonds within a certain distance cutoff, offers a natural way to obtain a more information-dense description. Specifically, our AuTopology tool uses the PyTorch Python library for implementing Message Passing Neural Networks (MPNNs), a type of GCNN. Nodes (atoms) are featurized by atomic number, valence, etc. while edges

(bonds) are featurized by distance, bond type, etc. Thus, atoms and bonds have some initial feature vector representation that is independent of their larger environment, while the connectivity of the molecule is represented by an adjacency matrix.

Atomic feature vectors, $\vec{r}_i$, are iteratively updated via a message-passing scheme:

$$\vec{r}_i^{(t+1)} = U\left(\vec{m}_i^{(t)}\right) \tag{6}$$

$$\vec{m}_i^{(t)} = \sum_{j \in \mathcal{N}(i)} M\left(\vec{\Omega}_{ij}^{(t)}\right) \tag{7}$$

where superscripts represent the convolution step, $\mathcal{N}(i)$ are the bonded neighbors of atom $i$, $U$ and $M$ are neural network update and message functions, $\vec{m}_i$ is the cumulative message sent to atom $i$ from its bonded neighbors, and $\vec{\Omega}_{ij}$ is a vector representing the interaction of atoms $i$ and $j$, usually a concatenation of their atomic feature vectors and the feature vector of the intervening bond. A schematic representation of the message-passing approach is given in the figure below.

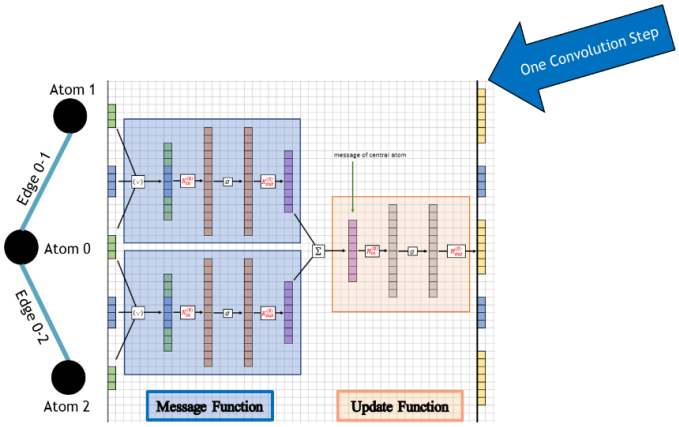


**Figure 2.** One convolution step of a message passing neural network

After $R$ convolutions, each updated atomic feature vector incorporates information from all atoms and bonds in its radius-$R$ chemical neighborhood. It has been proven that MPNNs are capable of learning injective neighborhood mappings such that there is a one-to-one correspondence between learned atom embeddings and the radius-$R$ neighborhoods present in the training data [12]. In principle, this would be equivalent to learning the categorical labels that are used to distinguish atoms and bonds in AMBER, CHARMM, and the other modern FF packages. Given a set of molecular geometries and their *ab initio*-calculated energies, there are two paths to proceed.

1. Directly fit the FF parameters (equilibrium bond lengths, spring constants, Lennard-Jones parameters, etc.)
2. Use the learned atom embeddings to predict FF parameters via auxiliary parameter-predictor networks (one for equilibrium bond length, one for spring constant, etc.)

The first option is given simply to point out why this method, which has been attempted by other research groups [51], is inherently limited. Because parameters are fit directly, there is no driving force for the atom embeddings to extract any useful information from the chemical environment beyond a simple unique label. Such a model would not be able to address a chemical neighborhood not seen in the training data, and would therefore have zero transferability. Our unique approach is to follow the second route, in which the MPNN layers applied to the molecular graph are forced to learn informative neighborhood representations, much like the continuous spatial kernels of neural network potentials but without the extreme over-parameterization that can lead to their failure to interpolate. The node embeddings should contain the physics necessary to calculate all FF parameters for the interactions in which they are involved.

As a concrete example, the bond between atoms $i$ and $j$, with MPNN-learned embeddings $\vec{r}_i^{(R)}$ and $\vec{r}_j^{(R)}$, will be parameterized as

$$r_{ij}^{eq} = \text{bondnet\_req}\left(\vec{r}_i^{(R)} \oplus \vec{r}_j^{(R)}\right) \tag{8}$$

$$k_{ij} = \text{bondnet\_k}\left(\vec{r}_i^{(R)} \oplus \vec{r}_j^{(R)}\right) \tag{9}$$

where $\oplus$ is any permutation invariant function (such as addition) and bondnet_req and bondnet_k are feed-forward neural networks that are the same for all bonds. The same procedure will be carried out for all angles, dihedrals, improper torsions, and non-bonded interactions of the molecular graph, with separate neural networks for each parameter or group of parameters.

Lastly, it remains to decide what terms will be included in the energy expansion (Eq. 1) and what their functional forms should be. The previous sections have outlined the principles of energy partitioning, which guide the choice of energy terms from a quantum mechanics-based, bottom-up perspective, as well as the more chemical intuition-driven top-down approach in which certain energetic contributions are only considered explicitly if needed, such as polarizability. AuTopology allows the user to treat the different underlying in a modular, “lego-like” way, so that the importance of each term can be explored systematically as various contributions are turned on and off.

# Methods

## Data Generation and Management

A crucial requirement is that the ML models be trained on data that densely samples chemical configuration space. This means that (1) the training data should be chemically diverse, so that each type of atom (carbon, oxygen, …) is seen in a wide range of chemical contexts, both in terms of its bonded and non-bonded interactions, and (2) configurationally diverse. Requirement (1) is needed to provide a driving force for the MPNN layers to extract relevant physics from chemical neighborhoods, while requirement (2) is needed to avoid under-determined models, in which the training data permit multiple equivalently accurate parameterizations. The notion of chemical

diversity at fixed configuration is illustrated in the figure below, which shows (excluding hydrogens) all possible 11-atom chains containing at least two non-adjacent oxygens.

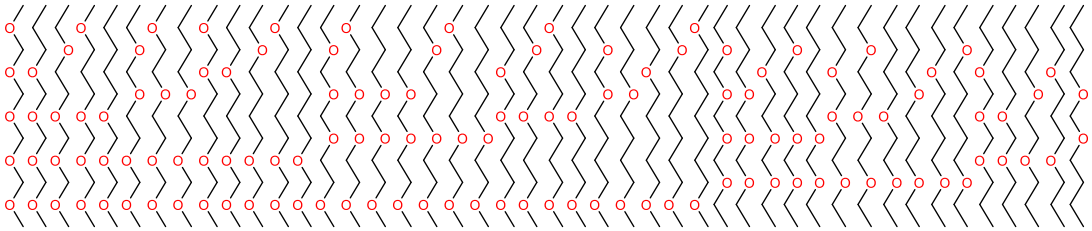

**Figure 3.** The CCS must be densely sampled for the MPNN to extract physically relevant information.

Machine learning models are written in Python using the PyTorch library for accelerated execution of training on CUDA-compatible GPUs. Trained force fields are implemented in LAMMPS for executing molecular dynamics simulations of bulk systems and for calculating thermodynamic and kinetic material properties. A set of PostgreSQL relational databases using the Django framework is used to store the results of quantum chemistry calculations and for generating ML featurizations of that data for training. Chemical objects stored in the database include molecular graphs, geometries, interacting molecular clusters, and reacting systems. A job submission system is also used to launch calculations from the database, permitting an automated platform for generating and training new batches of data in a high-throughout manner. The pipeline of chemistry → molecular graph → database → training of ML model(s) is shown in the figure below.

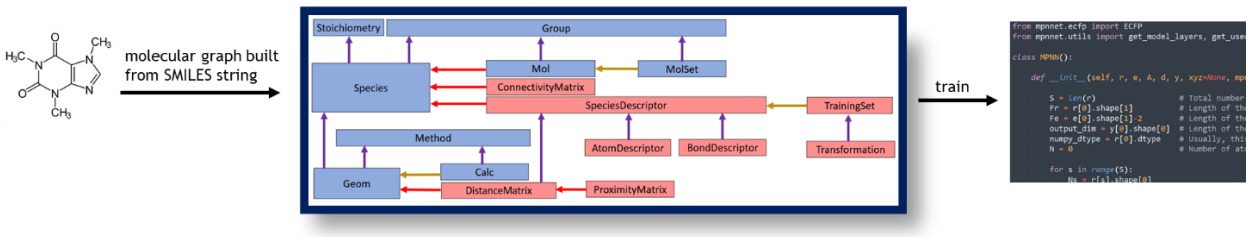


**Figure 4.** A high throughput virtual screening (HTVS) platform for data generation and training

## Training

We initially restrict the functional form of our force field to harmonic bond, angle, and improper dihedral energy terms with OPLS-style proper dihedrals for bonded interactions. For long-range (that is, non-bonded) interactions, we employ fixed atom-centered Coulombic point charges and Lennard-Jones dispersion interactions with geometric mixing of parameters. As with the OPLS force field, we include 1-4 non-bonded interactions scaled by a factor of 0.5. This simple functional form is not expected to achieve chemical accuracy even when fit to narrow regions of chemical space, much less the large diversity of chemistries present in the ANI-1 data set containing ~57K small organic molecules. However, our present aim is not to fully develop a universal force field with continuous atom typing, but rather to provide a proof of principle of a novel hybrid-ML-and

classical-FF method by demonstrating the feasibility of machine learning methods to interpolate force field parameters across chemical space.

A brief description of our two benchmark datasets follows. The ANI-1 dataset is a collection of ~20M molecular conformations, provided as XYZ files, spanning 57,462 C, H, O, and N-containing species with up to 8 heavy atoms. The energy of each geometry was calculated at the ωB97x level of theory and each species is labeled by its SMILES string. The geometries present in the ANI-1 dataset are sampled using normal node displacement about the calculated equilibrium structure, with the number of geometries per species chosen such that larger molecules are sampled less frequently than smaller ones. This choice avoids the problem of over-representing certain molecular graph substructures in the training data and, therefore, biasing the resulting force field to those chemical motifs.

While ANI-1 is designed to cover a wide region of chemical space with relatively few conformations per species, the MD-17 dataset contains ~1M geometries spanning only 8 small organic molecules: benzene, uracil, naphthalene, aspirin, salicylic acid, malonaldehyde, ethanol, and toluene. These geometries are obtained by running single-molecule MD trajectories at 500 K using the CCSD(T) level of theory. Furthermore, unlike ANI-1, both forces and energies are provided for each geometry.

Continuous atomic embeddings are trained using a message passing neural network as implemented in the Chemprop package. SMILES strings are converted to molecular graphs, with bonds and atoms labeled with a standard set of chemical features provided by the RDKit package. Because the RDKit atomic ordering does not match those given in the XYZs provided in the ANI-1 dataset, the xyz2mol package is used to determine the adjacency matrix of each species from its minimum-energy conformation in the training data using a standard set of distance thresholds to determine chemical bonding.

The RDKit bond and atom features include atomic number, aromaticity, degree, valence, ring membership, and bond order. While these features alone are enough to infer some characteristics of an atom's chemical environment, they do not provide the one-to-one mappings that are necessary for atom typing in existing classical force fields. Therefore, one or more MPNN layers are required to learn sufficiently descriptive atom embeddings. During each convolution, the feature vectors $\vec{r}_n$ of each atom's neighbor $n$ is concatenated with the feature vector $\vec{b}_n$ of the associated bond and sent through two nonlinear layers to obtain a message $\vec{m}_n = M\left[\text{cat}\left(\vec{r}_n, \vec{b}_n\right)\right]$. These messages are sum-pooled over the central atom's neighbors before being sent through another neural network, $U[\sum_n \vec{m}_n]$, yielding an updated atomic feature vector. This is done one or more times corresponding to the desired radius of the MPNN convolutional layer.

As described previously, the atomic embeddings that result from the MPNN layers are used to obtain continuous representations of each of the topologies present in the molecule graph (Eqs. 8 and 9). These representations must be invariant with respect to permutation of the atomic indices. For bonds, this requirement is satisfied by simply setting the bond feature vector as a sum of the

two neighboring atom embeddings. That is, $\vec{r}_{A-B} = \vec{r}_A + \vec{r}_B$. For angles, the central atom embedding is concatenated with the sum of the terminal atom embeddings: $\vec{r}_{A-B-C} = \text{cat}(\vec{r}_B, \vec{r}_A + \vec{r}_C)$. Special care must be taken when choosing how to represent dihedral topologies. For dihedral A-B-C-D, we distinguish between the internal atoms B and C and their associated terminal atoms A and D. The embedding of the dihedral as a whole is chosen as $\vec{r}_{A-B-C-D} = f[\text{cat}(\vec{r}_B, \vec{r}_A)] + f[\text{cat}(\vec{r}_C, \vec{r}_D)]$, where $f$ is a learnable nonlinear layer. For prediction of atom-specific properties like charges and LJ parameters, the MPNN embeddings are used directly.

To handle the calculation of these topologies and their latent space representations in an efficient, GPU-scalable way, we developed the Pytorch-based AuTopology package. Given one or more chemical species with one or more geometries per species, the AuTopology tool creates memory-efficient graph datasets that keep track of bonds, angles, dihedral and other topologies, their geometric properties (such as bond lengths, torsion angles, and their derivatives), and their embeddings, as was shown in Fig. 1. Given these geometries, AuTopology then determines conformation energies and, optionally, per-atom forces. Energies are determined using an analytical expression of the desired force field (such as OPLS or COMPASS). Forces can be calculated analytically or using automatic differentiation of the previously calculated energy. The atomic embeddings that go into the parameter predictors (BondNet, AngleNet, DihedralNet, etc.) come directly from MPNN performed on the molecular graphs by the Chemprop package, so that the calculation of forces from the input molecular graphs is end-to-end differentiable.

## Results

We first demonstrate the ability to interpolate in conformational space using the MD-17 benchmark dataset with an OPLS functional form for the force field. Note that in this case, since we do not require interpolation in chemical space, the initial MPNN layer is replaced with a simple assignment of one-hot atom types based on each atom's radius 2 molecular graph environment. Training is performed by minimizing the RMSE of atomic forces and energies for 200 configurations of toluene, with a resulting RMSE of 7.2 kcal/mol- Å for forces and 1.5 kcal/mol for energies. A comparison of target and predicted atomic force components and energies are shown in Fig. 5, where green predictions are those within one standard deviation of the target. Similarly, Fig. 6 shows the results of training on forces for 1000 geometries spanning the 8 MD-17 species, yielding an RMSE of 7.2 kcal/mol-Å. In the latter case, $R = 4$.

These results exceed the accuracy that one would obtain by simply using off-the-shelf parameters provided by the OPLS force field. This is because OPLS was optimized to be "good enough" for a wide range of chemistries, without being tailored to any particular region of the CCS. Indeed, the AuTopology tool is, in general, capable of generating parameterizations that lead to force and energy fittings whose accuracies exceed those of the corresponding published CFFs. In other words, using this tool, force fields can be trained that are as specific as needed for a desired application.

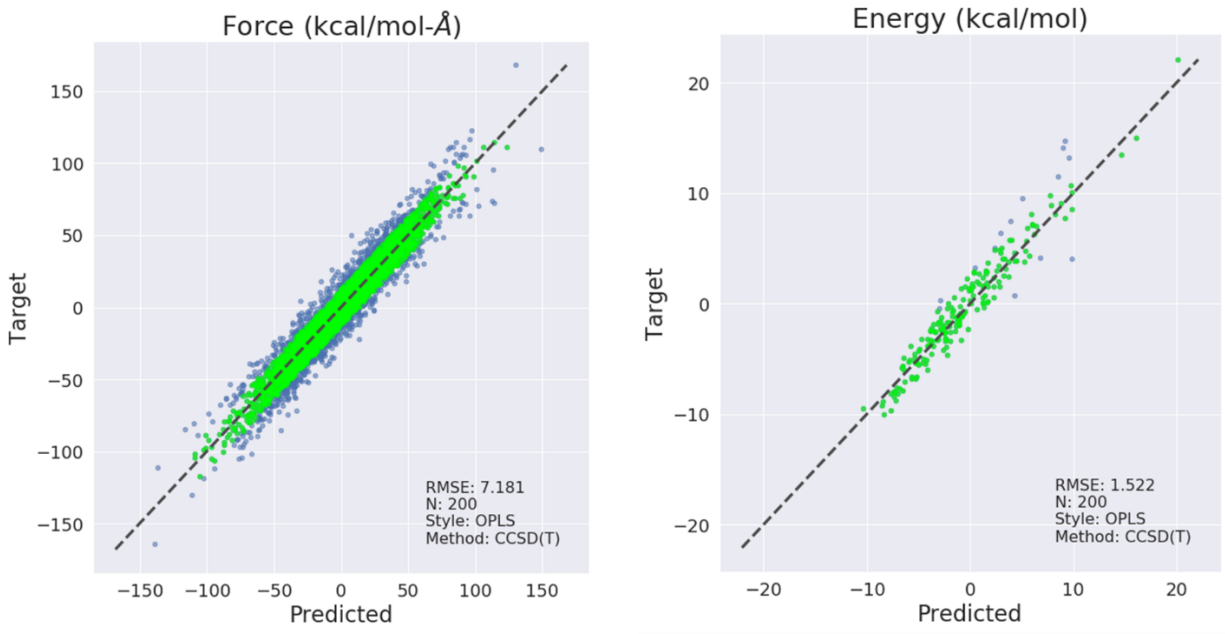


**Figure 5.** Predictions of an AuTopology-trained OPLS-style force field for toluene from the MD-17 dataset

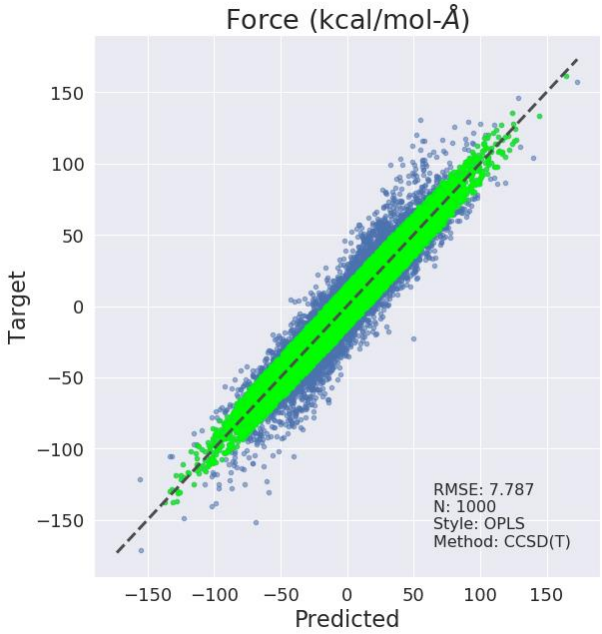


**Figure 6.** Predictions of an AuTopology-trained OPLS-style force field for all 8 species of the MD-17 dataset

We next take a closer look at the convergence of the individual force field parameters. For this example, we look at conformations of a cluster of Li+, TFSI-, and a boron-containing species denoted by the smiles string CCB1C2CCCC1CCC2. By investigating these cluster, as opposed to isolated species, we are able to better explore the sensitivity of the accuracy of the force field to specific long-range interactions. Fig. 7 shows the training results for a set of 30 conformations of the [Li+].[TFSI-].CCB1C2CCCC1CCC2 cluster. The structure of the boron-containing species is also shown. The data points have been colored by atom type while the loss curves have been colored by the individual species.

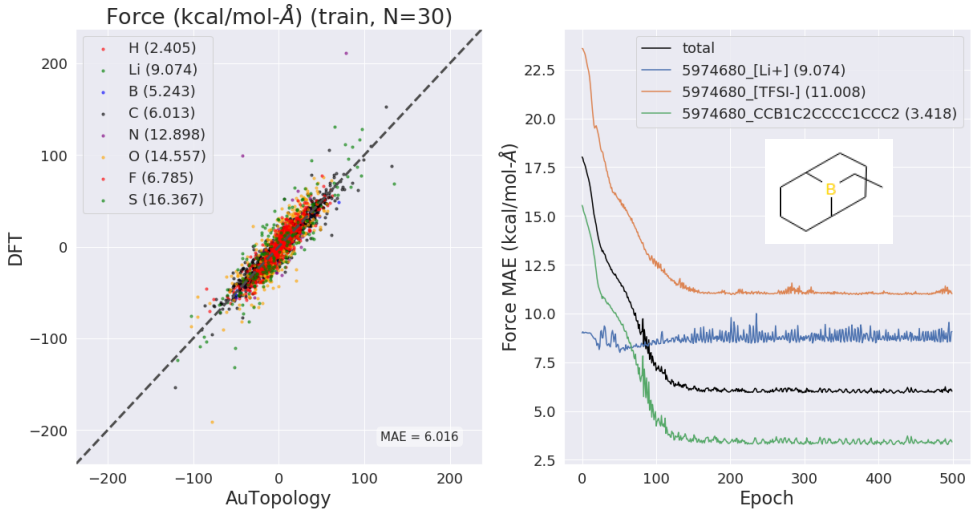


**Figure 7.** Atom-specific force predictions and species-specific loss curves of an AuTopology-trained OPLS-style force field for [Li+].[TFSI-].CCB1C2CCCC1CCC2 clusters. The numbers in parenthesis are final force MAE values for atoms and species, respectively, where 5974680 is simply a database ID for the given cluster. The structure of the boron-containing species is shown in the inset.

It is immediately clear that the intramolecular interactions of the boron-containing species show good accuracy, with a force MAE of 3.4 kcal/mol- Å. Checking the convergence of the long-range parameters (LJ sigma and epsilon and Coulombic partial charges), as shown in Fig. 8, we see unphysical values for the excluded volume, with a maximum value of around 12 Angstroms for nitrogen. Moreover, with no further regularization term added to restrict the total charges, we see overestimation of the partial charge of the lithium ion and epsilon values that saturate at an arbitrarily chosen upper limit of 0.3 kcal/mol.

This serves to demonstrate how the AuTopology tool can be used in practice. Because all FF parameters are learned jointly, it is possible to manually modify or otherwise influence the training of certain parameters, while allowing the other parameters of the now-constrained FF to compensate. In this example, it was found that the long-range interactions are sufficiently weak in the clusters used for training that the associated parameters, such as LJ epsilon and sigma, can fluctuate to unreasonable or unphysical values without significant loss in the overall fit to the training data, even as the bonded parameters do not show similar pathologies (see Fig. 9).

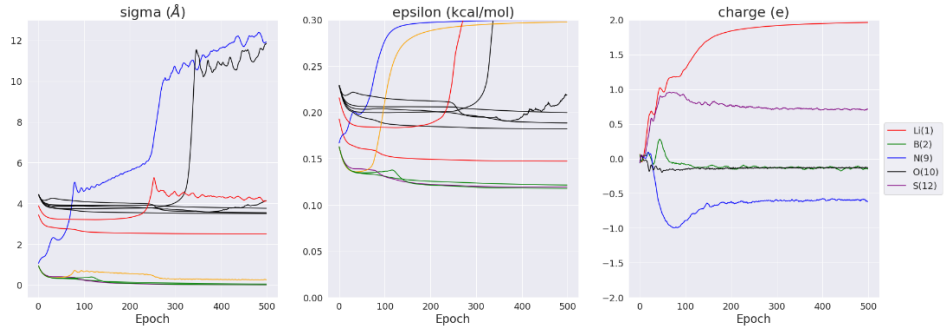


**Figure 8.** Convergence of long-range parameters (LJ sigma and epsilon and Coulombic charges) showing that unphysical values present among FF parameters do not always correspond to low-accuracy force predictions

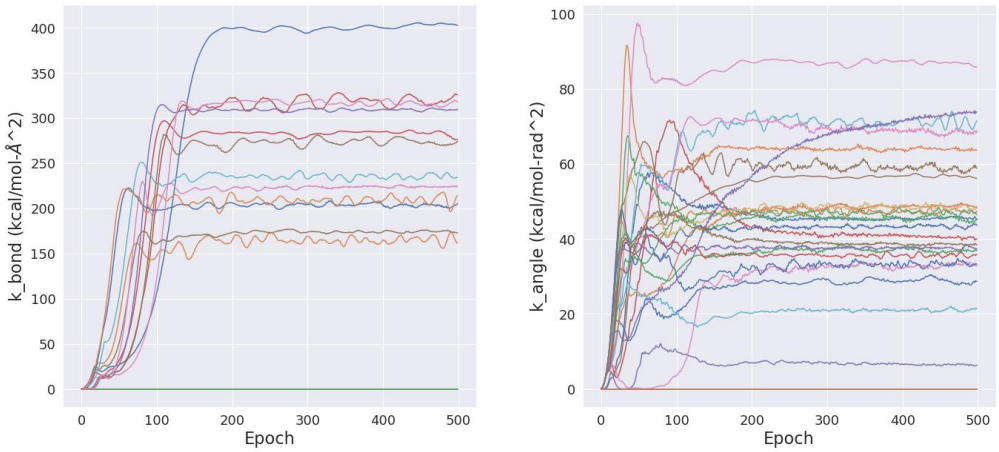


**Figure 9.** Convergence of bond and angle constants show physically reasonable magnitudes even as long-range parameters are unphysical.

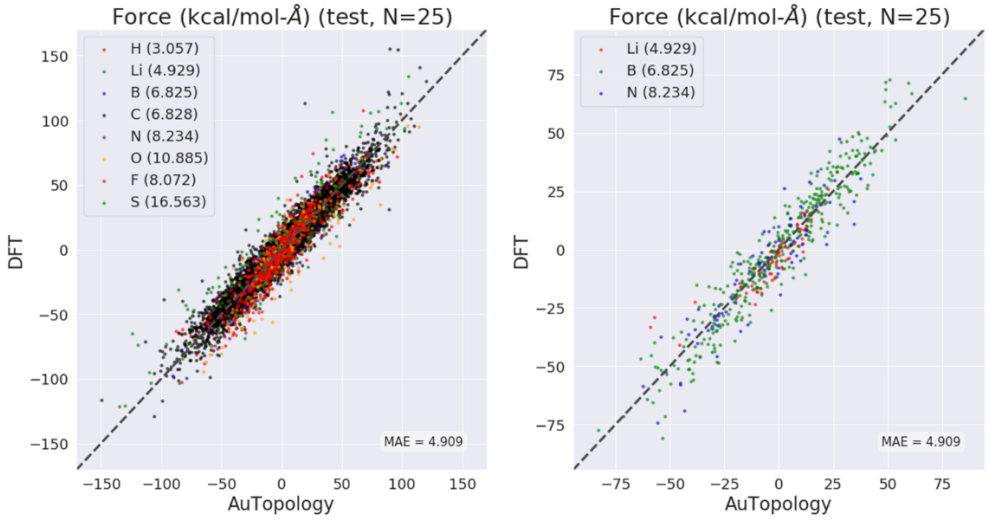


**Figure 10.** Simple tricks during training, such as including a loss term that favors LJ sigma values that are as large as possible while putting a floor on the associated LJ epsilon values, yields improved FF fitting and physically reasonable parameter magnitudes.

However, by simply putting a floor on the possible epsilon values and incentivizing large sigma values, a more accurate FF was trained whose long-range parameters are physically reasonable, as shown in Fig. 10. In general, the AuTopology tool is well-equipped to handle such exploration of the underlying physics of a problem, and can readily be extended by future users to include novel force field terms and interactions.

## Summary and Conclusion

Despite the rapidly expanding computational power available to modern researchers, it will remain infeasible to perform high-accuracy *ab initio* simulations for large systems of interest for the foreseeable future. Therefore, computationally cheaper force fields are required. At present, classical force fields such as AMBER, CHARMM, and AMOEBA suffer from limited transferability between materials classes because their functional forms do not faithfully represent the underlying physics. On the other extreme, recently developed neural network potentials can fit molecular potential energy surfaces with high accuracy, but their flexible functional forms make it difficult to extract chemical insights from their predictions. Moreover, they are susceptible to failure if used to extrapolate beyond their training data. We developed a tool, AuTopology, to systematically explore the intermediate space using graph-based machine learning methods to optimally parameterize force fields that incorporate physically realistic underlying functional forms, and showed several applications of its use to demonstrate its flexibility and utility.

# References


1. A. Amidi, S. Amidi, D. Vlachakis, V. Megalooikonomou, N. Paragios, and E. I. Zacharaki. EnzyNet: enzyme classification using 3D convolutional neural networks on spatial representation. arXiv:1707.06017, 2017.

2. W. Torng and R. B. Altman, Bioinformatics (2018).

3. Q. U. Ain, A. Aleksandrova, F. D. Roessler, and P. J. Ballester, Wiley Interdisciplinary Reviews: Computational Molecular Science 5, 405 (2015).

4. C. W. Coley, R. Barzilay, W. H. Green, T. S. Jaakkola, and K. F. Jensen, Journal of Chemical Information and Modeling 57, 1757 (2017).

5. T. Blaschke, M. Olivecrona, O. Engkvist, J. Bajorath, and H. Chen. Application of generative autoencoder in de novo molecular design. arXiv:1711.07839, 2017.

6. B. Sanchez-Lengeling and A. Aspuru-Guzik, Science 361, 360 (2018).

7. W. L. Hamilton, R. Ying, and J. Leskovec. Representation Learning on Graphs: Methods and Applications. arXiv:1709.05584, 2017.

8. R. Gomez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernandez-Lobato, B. Sanchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. arXiv:1610.02415, 2016.

9. C. Angermueller, T. Pärnamaa, L. Parts, and O. Stegle, Molecular Systems Biology 12, 878 (2016).

10. Y. Li, O. Vinyals, C. Dyer, R. Pascanu, and P. Battaglia. Learning Deep Generative Models of Graphs. arXiv:1803.03324, 2018.

11. K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, Nature Communications 8, 13890 (2017).

12. K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How Powerful are Graph Neural Networks? arXiv:1810.00826, 2018.

13. A. Grover, A. Zweig, and S. Ermon. Graphite: Iterative Generative Modeling of Graphs. arXiv:1803.10459, 2018.

14. J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural Message Passing for Quantum Chemistry. arXiv:1704.01212, 2017.

15. D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gomez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional Networks on Graphs for Learning Molecular Fingerprints. arXiv:1509.09292, 2015.

16. M. P. Allen and D. J. Tildesley, Computer Simulation of Liquids (Oxford University Press, 2017).

17. R. Drautz, Physical Review B 99, (2019).

18. L. Zhao, M. von Hopffgarten, D. M. Andrada, and G. Frenking, Wiley Interdisciplinary Reviews: Computational Molecular Science 8, e1345 (2017).

19. K. Morokuma, The Journal of Chemical Physics 55, 1236 (1971).

20. P. Maxwell, Á. M. Pendás, and P. L. A. Popelier, Physical Chemistry Chemical Physics 18, 20986 (2016).

21. M. Rafat and P. L. A. Popelier, Journal of Computational Chemistry 28, 292 (2007).

22. S. Cardamone, T. J. Hughes, and P. L. A. Popelier, Physical Chemistry Chemical Physics 16, 10367 (2014).

23. M. V. Ivanov, M. R. Talipov, and Q. K. Timerghazin, The Journal of Physical Chemistry A 119, 1422 (2015).

24. Mu, D. S. Kosov, and G. Stock, The Journal of Physical Chemistry B 107, 5064 (2003).

25. P. L. A. Popelier, in Modern Charge-Density Analysis (Springer Netherlands, 2011), pp. 505–526.

26. M. G. Darley and P. L. A. Popelier, The Journal of Physical Chemistry A 112, 12954 (2008).

27. W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman, Journal of the American Chemical Society 117, 5179 (1995).

28. B. R. Brooks, C. L. Brooks III, A. D. Mackerell Jr., L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, and M. Karplus, Journal of Computational Chemistry 30, 1545 (2009).

29. W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives, Journal of the American Chemical Society 118, 11225 (1996).

30. M. G. Darley and P. L. A. Popelier, The Journal of Physical Chemistry A 112, 12954 (2008).

31. M. Rafat and P. L. A. Popelier, The Journal of Chemical Physics 124, 144102 (2006).

32. K. Vanommeslaeghe and A. D. MacKerell Jr., Biochimica et Biophysica Acta (BBA) - General Subjects 1850, 861 (2015).

33. G. Császár, Wiley Interdisciplinary Reviews: Computational Molecular Science 2, 273 (2011).

34. P. E. M. Lopes, E. Harder, B. Roux, and A. D. Mackerell, in Multi-Scale Quantum Models for Biocatalysis (Springer Netherlands, 2009), pp. 219–257.

35. Warshel, M. Kato, and A. V. Pisliakov, Journal of Chemical Theory and Computation 3, 2034 (2007).

36. P. Jungwirth and D. J. Tobias, Chemical Reviews 106, 1259 (2006).

37. I. Leontyev and A. Stuchebrukhov, Physical Chemistry Chemical Physics 13, 2613 (2011).

38. F.-Y. Lin, P. E. M. Lopes, E. Harder, B. Roux, and A. D. MacKerell Jr., Journal of Chemical Information and Modeling 58, 993 (2018).

39. T. A. Halgren and W. Damm, Current Opinion in Structural Biology 11, 236 (2001).

40. Y. Ding, D. N. Bernardo, K. Krogh-Jespersen, and R. M. Levy, The Journal of Physical Chemistry 99, 11575 (1995).

41. K. Vanommeslaeghe and A. D. MacKerell Jr., Biochimica et Biophysica Acta (BBA) - General Subjects 1850, 861 (2015).

42. I. Vorobyov, V. M. Anisimov, S. Greene, R. M. Venable, A. Moser, R. W. Pastor, and A. D. MacKerell, Journal of Chemical Theory and Computation 3, 1120 (2007).

43. H. Yu and W. F. van Gunsteren, The Journal of Chemical Physics 121, 9549 (2004).

44. J. 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 Jr., M. Head-Gordon, G. N. I. Clark, M. E. Johnson, and T. Head-Gordon, The Journal of Physical Chemistry B 114, 2549 (2010).

45. S. Cardamone, T. J. Hughes, and P. L. A. Popelier, Physical Chemistry Chemical Physics 16, 10367 (2014).

46. K. A. Dill, S. Bromberg, K. Yue, H. S. Chan, K. M. Ftebig, D. P. Yee, and P. D. Thomas, Protein Science 4, 561 (2008).

47. P. Cieplak, F.-Y. Dupradeau, Y. Duan, and J. Wang, Journal of Physics: Condensed Matter 21, 333102 (2009).

48. J. Behler, The Journal of Chemical Physics 134, 74106 (2011).

49. J. S. Smith, O. Isayev, and A. E. Roitberg, Chemical Science 8, 3192 (2017).

50. K. T. Schütt, P. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, and K. Müller. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. arXiv:1706.08566, 2017.

51. H. Chan, B. Narayanan, M. J. Cherukara, F. G. Sen, K. Sasikumar, S. K. Gray, M. K. Y. Chan, and S. K. R. S. Sankaranarayanan, The Journal of Physical Chemistry C (2019).

52. T. P. Senftle, S. Hong, M. M. Islam, S. B. Kylasa, Y. Zheng, Y. K. Shin, C. Junkermeier, R. Engel-Herbert, M. J. Janik, H. M. Aktulga, T. Verstraelen, A. Grama, and A. C. T. van Duin, Npj Computational Materials 2, (2016).

53. G. Henkelman, B. P. Uberuaga, and H. Jónsson, The Journal of Chemical Physics 113, 9901 (2000).