
End-to-End Differentiable Molecular Mechanics Force Field Construction

Yuanqing Wang*, Josh Fass, and John D. Chodera[†]

Computational and Systems Biology Program

Sloan Kettering Institute

Memorial Sloan Kettering Cancer Center

New York, NY 10065

yuanqing.wang@choderalab.org

Abstract

Molecular mechanics (MM) potentials have long been a workhorse of computational chemistry. Leveraging accuracy and speed, these functional forms find use in a wide variety of applications from rapid virtual screening to detailed free energy calculations. Traditionally, MM potentials have relied on human-curated, inflexible, and poorly extensible discrete chemical perception rules (*atom types*) for applying parameters to molecules or biopolymers, making them difficult to optimize to fit quantum chemical or physical property data. Here, we propose an alternative approach that uses *graph nets* to perceive chemical environments, producing continuous atom embeddings from which valence and nonbonded parameters can be predicted using a feed-forward neural network. Since all stages are built using smooth functions, the entire process of chemical perception and parameter assignment is differentiable end-to-end with respect to model parameters, allowing new force fields to be easily constructed, extended, and applied to arbitrary molecules. We show that this approach has the capacity to reproduce legacy atom types and can be fit to MM and QM energies and forces, among other targets.

1 Introduction

Molecular mechanics force fields—physical models that abstract molecular systems as interacting point charges that separate the energy into atom (nonbonded), bond, angle, and torsion terms—have powered *in silico* modeling to provide key insights and quantitative predictions in all aspects of chemistry, from drug discovery to material sciences [1, 2, 3, 4, 5, 6, 7, 8, 9]. While recent work in quantum machine learning (QML) potentials has demonstrated how flexibility in functional forms and training strategies leads to increased accuracy [10], these methods are orders of magnitude slower than popular molecular mechanics potentials, since the learned high-dimensional function approximator must be used also in simulation. On the other hand, the simpler physical energy functions of molecular mechanics (MM) models are compatible with highly optimized implementations that can exploit a variety of hardware [2, 11, 12, 13, 14, 15], but rely on complex and inextensible legacy *atom typing schemes* for parameter assignment: First, a set of rules are used to classify atoms into *atom types* that must encode any information about the chemical environment that will be used by subsequent steps [16]. Next, bond, angle, and torsion types are determined by the composing atom types. Finally, the parameters attached to atoms, bonds, angles, and torsions are assigned according to a table of these parameter classes. As a result, atoms, bonds, angles, or torsions with distinct

*Alternative Address: Physiology, Biophysics, and System Biology Ph.D. Program, Weill Cornell Medical College, Cornell University, New York, NY 10065; M.F.A. Program in Creative Writing, Division of Humanities and Arts, the City College of the City University of New York, New York, NY 10031.

[†]Corresponding Author. Email: john.chodera@choderalab.org

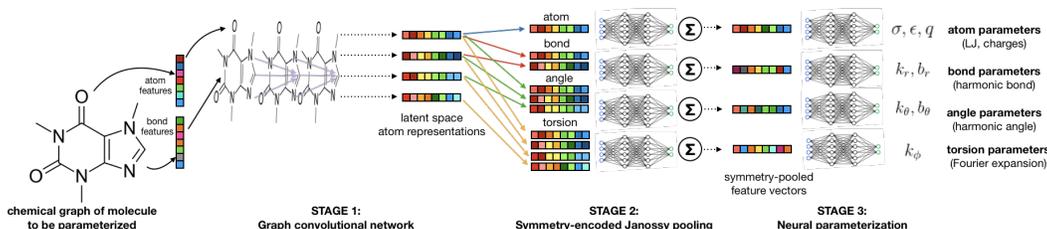


Figure 1: End-to-end differentiable assignment of parameters for a molecular mechanics potential energy function. In Stage 1, a graph net is used to generate latent atom embeddings describing local chemical environments from the chemical graph. In Stage 2, these atom embeddings are transformed into symmetry-encoded feature vectors for atom, bond, angle, and torsion inference via Janossy pooling. In Stage 3, molecular mechanics parameters are directly predicted from these feature vectors using feed-forward neural nets. This process is performed once per molecular species, allowing the potential energy to be rapidly computed using standard molecular mechanics implementations thereafter.

chemical environments that happen to fall into the same expert-derived category are forced to share a same set of parameters and lead to poor accuracy, while the explosion of discrete parameter classes describing equivalent chemical environments can lead to overfitting. Even with modern optimization frameworks [17, 18, 19] and sufficient data, parameter optimization is only possible in the continuous parameter space defined by these fixed atom types, while the mixed discrete-continuous optimization problem—jointly optimizing types and parameters—is intractable.

Here, we demonstrate a continuous alternative to discrete atom typing schemes that permits end-to-end differentiable optimization of both “typing” and parameter assignment, allowing the entire force field to be built, extended, and applied using standard machine learning frameworks. We hypothesize that graph neural networks (graph nets) have at least equivalent expressiveness with expert-derived typing rules. We first provide experimental evidence of this hypothesis by showing that, with acceptable errors: (1) graph nets can recover legacy atom types in a supervised classification task; (2) graph nets, when combined with appropriate pooling functions and a subsequent prediction stage, can learn MM energy function parameters when trained to fit the corresponding MM energies and forces. Next, we demonstrate the utility of such a model (which we call the *Extendable Surrogate Potential Optimized by Message-passing Algorithms*, or **Espaloma**) to construct end-to-end optimizable force fields with continuous atom types that can be used to fit quantum mechanics (QM) energies via auto-differentiating frameworks.

2 Theory

2.1 An end-to-end differentiable approach to molecular mechanics parameter assignment

Here, we show how our proposed framework, **Espaloma** (Figure 1), operates analogously to legacy force field typing schemes to generate MM parameters Φ_{FF} from a molecular graph \mathcal{G} and neural parameters \mathbf{W}_{NN} , $\Phi_{\text{FF}} \leftarrow \text{Espaloma}(\mathcal{G}, \mathbf{W}_{\text{NN}})$. We target the MM functional form (see SI 12.1):

$$\begin{aligned}
 U(\mathbf{x}; \Phi_{\text{FF}}) &= \sum_{i < j} 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r} \right)^{12} - \left(\frac{\sigma_{ij}}{r} \right)^6 \right] + C \frac{q_i q_j}{r} \\
 &+ \sum_{\text{bonds}} \frac{k_r}{2} (r - r_0)^2 + \sum_{\text{angles}} \frac{k_\theta}{2} (\theta - \theta_0)^2 + \sum_{\text{torsions}} \sum_n k_{\phi, n} \cos[n(\phi - \phi_n)]. \quad (1)
 \end{aligned}$$

Stage 1: Graph nets generate a continuous atom embedding, replacing legacy atom typing. Our proposed scheme uses graph nets to perceive the chemical environments of atoms from a chemical graph, generating continuous embeddings $h_v \in \mathbb{R}^{|\mathcal{G}| \times D}$ for each atom v . For an introduction to graph nets in the context of molecular modeling, see SI 12.2.

Stage 2: Symmetry-encoded pooling generates continuous bond, angle, and torsion embeddings, replacing discrete types. To produce representations of bond, angle, and torsion environments that are expressive and invariant w.r.t. the ordering of atoms, we use Janossy pooling [20] to

derive their embeddings h_r, h_θ, h_ϕ from atom embeddings h_v ,

$$h_{r_{ij}} = \text{NN}_r([h_{v_i} : h_{v_j}]) + \text{NN}_r([h_{v_j} : h_{v_i}]); \quad (2)$$

$$h_{\theta_{ijk}} = \text{NN}_\theta([h_{v_i} : h_{v_j} : h_{v_k}]) + \text{NN}_\theta([h_{v_k} : h_{v_j} : h_{v_i}]); \quad (3)$$

$$h_{\phi_{ijkl}} = \text{NN}_\phi([h_{v_i} : h_{v_j} : h_{v_k} : h_{v_l}]) + \text{NN}_\phi([h_{v_l} : h_{v_k} : h_{v_j} : h_{v_i}]), \quad (4)$$

where columns $(\cdot : \cdot)$ denote concatenation.

Stage 3: Neural parametrization of atoms, bonds, angles, and torsions replaces tabulated parameterization. Finally, feed-forward neural networks learn the mapping from atom, bond, angle, and torsion embeddings to MM parameters Φ_{FF} :

$$\{\epsilon_v, \sigma_v\} = \text{NN}_{v_{\text{readout}}}(h_v); \{k_r, b_r\} = \text{NN}_{r_{\text{readout}}}(h_r); \{k_\theta, b_\theta\} = \text{NN}_{\theta_{\text{readout}}}(h_\theta); \{k_\phi\} = \text{NN}_{\phi_{\text{readout}}}(h_\phi), \quad (5)$$

Partial atomic charges can also be determined from the molecular graph in a fully differentiable manner using a graph net that predicts physical parameters for a charge-equilibration model [21].

2.2 Training and inference

While traditional force fields based on discrete atom types are only differentiable in the molecular mechanics parameters they assign, our model is fully differentiable in all model parameters W_{NN} that govern both the assignment of atom embeddings h_v , and subsequent assignment of MM parameters Φ_{FF} . We can therefore use gradient-based optimization to tune all of these parameters to fit arbitrary differentiable target functions. Examples include classification loss (as in Section 3.1), parameter regression loss, and likelihoods incorporating any differentiable observables (as in Section 3.2 and 3) such as energies, forces, and physical properties (e.g. densities, dielectric constants, and free energies).

When it comes to deployment, in contrast with many QML force fields (e.g. [10, 22, 23]), a neural model does not have to be executed at each time step during simulation. Once the model is trained, the MM parameters the model generates Φ_{FF} can be seamlessly ported to optimized MM packages [2, 11, 12, 13], providing the same speed as traditional force fields.

3 Experiments

Traditional molecular mechanics force field parameter assignment (e.g., [24, 25, 26, 27]) uses the attributes of atoms and their neighbors (such as atomic number, hybridization, and aromaticity) to assign discrete atom types, and then assigns atom, bond, angle, and torsion parameters based on these discrete types. Here, we not only demonstrate that graph nets can learn these legacy atom type assignments with high accuracy, but they can also be trained directly on energies and forces to learn both typing and parameter assignment simultaneously. For code availability and experimental details, see SI 6 and 7.

3.1 Graph nets have the capacity to reproduce legacy atom typing with high accuracy.

Assigning discrete atom types closely resembles a two- or three-step Weisfeiler-Leman test [28], which has been shown to be approximated by some graph neural network architectures [29]. Here, we ask whether graph nets have at least equivalent expressiveness with legacy atom typing schemes by training them to reproduce legacy (human-designed) atom types for a general small organic molecule force field, parm@frosst [30]. We randomly selected a subset (1000 molecules) of ZINC validation dataset [31] provided with parm@frosst to validate atom typing implementations [30] to use for training and validation tasks. As is shown in Figure 2, graph nets reproduce legacy atom types with high accuracy, with discrepancies occurring at sites where typing rules are less unambiguous and where samples are scarce.

3.2 Espaloma recovers MM parameters and energies with high fidelity

We next assess how accurately Espaloma could learn to reproduce MM parameters from end-to-end training directly on potential energies of snapshots from an atom type based MM force field, and how well this approach could learn to generalize. In order to focus on the *functional* complexity of

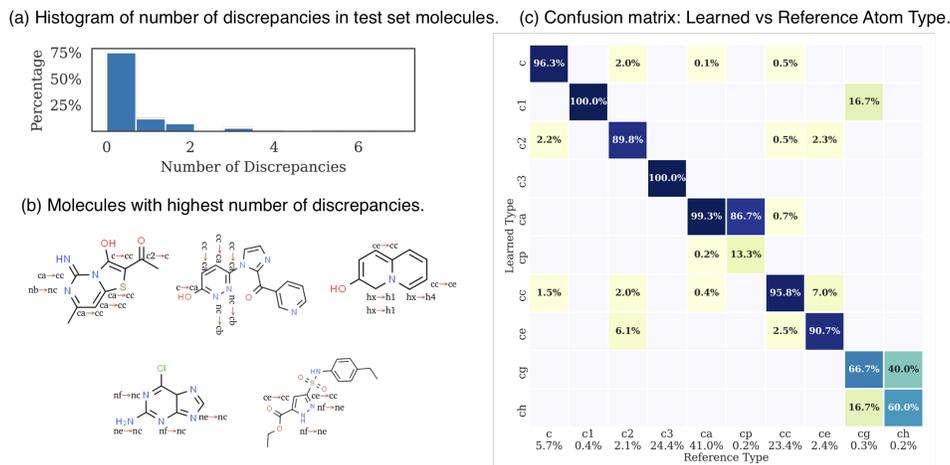


Figure 2: Graph nets can reproduce legacy atom types with high accuracy (98.31%^{98.63%}). (a) Histogram of number of discrepancies in molecules within the test dataset; (b) Illustrations of molecules in the test dataset with highest numbers of discrepancies between graph net-assigned and reference atom types; (c) Distribution of predicted atom types for each reference atom type; on-diagonal values indicate agreement. The percentages annotated under x-axis denote the relative abundance within the test dataset. Only the common carbon types are included in the confusion matrix here; for full confusion matrix across all atom types, see SI Figure 4.

the MM potentials and limit the scope of *chemical diversity*, we choose a minimal dataset of small alkanes, ethers, and alcohols with non-aromatic rings from AlkEthOH dataset [32], and generate a series of configuration snapshots using short high-temperature MD (further details in section 7). We report the test set performance in Table 1.

Quantity	RMSE	Test MAPE	R^2	RMSE	Training MAPE	R^2
Harmonic Bond + Angle Energy (kcal/mol)	0.4392 ^{0.3392} _{0.4392}	0.0157 ^{0.0162} _{0.0153}	0.9958 ^{0.9961} _{0.9955}	0.7726 ^{0.7958} _{0.7496}	0.0277 ^{0.0285} _{0.0269}	0.9953 ^{0.9957} _{0.9950}
Bond Force Constant k_r (kcal / (mol * angstrom ** 2))	35.4048 ^{30.2660} _{18.0387}	0.0180 ^{0.0215} _{0.0148}	0.8619 ^{0.9693} _{0.7154}	57.1240 ^{64.8255} _{49.2791}	0.0299 ^{0.0330} _{0.0271}	0.7391 ^{0.8095} _{0.6466}
Equilibrium Bond Length b_r (angstrom)	0.0127 ^{0.0200} _{0.0013}	0.0015 ^{0.0021} _{0.0011}	0.9956 ^{1.0000} _{0.9890}	0.0135 ^{0.0155} _{0.0117}	0.0299 ^{0.0330} _{0.0270}	0.7391 ^{0.8111} _{0.6590}
Angle Force Constant k_θ (kcal / (mol * rad ** 2))	3.7995 ^{3.9648} _{3.6293}	0.0276 ^{0.0290} _{0.0264}	0.8601 ^{0.8805} _{0.8361}	44.4132 ^{51.4202} _{36.3418}	0.0464 ^{0.0492} _{0.0436}	-0.2451 ^{-0.1895} _{-0.2839}
Equilibrium Angle Value b_θ (rad)	0.0043 ^{0.0045} _{0.0041}	0.0018 ^{0.0015} _{0.0017}	0.9202 ^{0.9018} _{0.9335}	0.0480 ^{0.0558} _{0.0397}	0.0059 ^{0.0070} _{0.0049}	0.4902 ^{0.5411} _{0.4420}

Table 1: Espaloma recovers MM energies and parameters when fit to MM energies. RMSE, MAPE, and R^2 between reference and predicted MM energies and parameters. Note that MAPE (Mean Absolute Percentage Error) reports a fraction, rather than a percentage.

3.3 Espaloma can fit QM energies directly to build new MM force fields

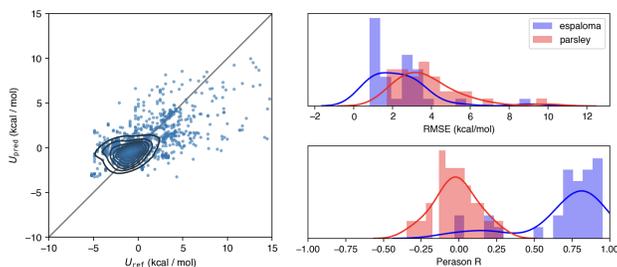


Figure 3: Espaloma can be directly optimized against QM energies to produce a new MM force field with comparable accuracy to traditional MM force field fitting schemes. Overall test set RMSE: 3.4966^{4.0498}_{2.9526} kcal/mol, compared to 3.7365^{4.4559}_{3.0209} kcal/mol for Parsley force field [16, 33]. Both reference and predicted energy are centered to have zero mean for each molecule.

Finally, we repeat the end-to-end fitting experiment (Section 3.2) directly using a quantum chemical (QM) target used to train MM force fields—the energies and forces in an Open Force Field [34] optimization dataset 1.0 in QCArchive [35]. We selected chemical species with more than 1000 snapshots, and randomly choose 1000 snapshots within each system. The test set performance is

reported in Figure 3. Since nonbonded terms are generally optimized to fit other condensed-phase properties, we focus here on optimizing valence parameters to fit gas-phase QM, fixing the nonbonded energies using a legacy force field [34]. According to the measures reported here, Espaloma approach can construct MM force fields with QM-fitting performance comparable to force fields based on discrete atom-typing.

4 Discussion

Here, we have demonstrated that graph nets have the capacity to, with high accuracy, reproduce legacy atom typing, fit a traditional molecular mechanics force field, generalize it to new molecules, and learn new force fields directly from quantum chemical energies and forces. The neural framework also affords the modularity to include more types of parameters (and thereby extended functional forms, such as Class II force fields [36]), while making it easy to refit the entire force field afterwards.

5 Broader impact

This work reduces technical barriers to data-driven force field parameterization. The most important immediate applications are seen in drug discovery, where it is valuable to tailor force fields rapidly to data in a lead series. If successful, the proposed methods will make it cheaper, easier, and faster to fit force fields to data using open source software and less human effort. This could disrupt the profit models of commercial force field vendors. (Also, unauthorized reverse-engineering of commercial force fields could become more of a concern.) Improved capabilities for fitting MM models to data could magnify the competitive advantages that firms can gain from in-house private data sources.

References

- [1] J. W. Ponder and D. A. Case, "Force fields for protein simulations," in *Advances in protein chemistry*, vol. 66, pp. 27–85, Elsevier, 2003.
- [2] D. Van Der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, and H. J. Berendsen, "Gromacs: fast, flexible, and free," *Journal of computational chemistry*, vol. 26, no. 16, pp. 1701–1718, 2005.
- [3] D. A. Case, T. E. Cheatham III, T. Darden, H. Gohlke, R. Luo, K. M. Merz Jr, A. Onufriev, C. Simmerling, B. Wang, and R. J. Woods, "The amber biomolecular simulation programs," *Journal of computational chemistry*, vol. 26, no. 16, pp. 1668–1688, 2005.
- [4] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale, and K. Schulten, "Scalable molecular dynamics with namd," *Journal of computational chemistry*, vol. 26, no. 16, pp. 1781–1802, 2005.
- [5] F. E. Calculations, "Theory and applications in chemistry and biology," *Springer Series in Chemical Physics*, vol. 86, 2007.
- [6] C. Wang, G. Pilania, S. Boggs, S. Kumar, C. Breneman, and R. Ramprasad, "Computational strategies for polymer dielectrics design," *Polymer*, vol. 55, no. 4, pp. 979 – 988, 2014.
- [7] C. Li and A. Strachan, "Molecular scale simulations on thermoset polymers: A review," *Journal of Polymer Science Part B: Polymer Physics*, vol. 53, no. 2, pp. 103–122, 2015.
- [8] H. Sun, Z. Jin, C. Yang, R. L. Akkermans, S. H. Robertson, N. A. Spenley, S. Miller, and S. M. Todd, "Compass ii: extended coverage for polymer and drug-like molecule databases," *Journal of molecular modeling*, vol. 22, no. 2, p. 47, 2016.
- [9] E. Harder, W. Damm, J. Maple, C. Wu, M. Reboul, J. Y. Xiang, L. Wang, D. Lupyan, M. K. Dahlgren, J. L. Knight, *et al.*, "Opls3: a force field providing broad coverage of drug-like small molecules and proteins," *Journal of chemical theory and computation*, vol. 12, no. 1, pp. 281–296, 2016.
- [10] J. S. Smith, O. Isayev, and A. E. Roitberg, "Ani-1: an extensible neural network potential with dft accuracy at force field computational cost," *Chemical science*, vol. 8, no. 4, pp. 3192–3203, 2017.

- [11] P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp, L.-P. Wang, A. C. Simmonett, M. P. Harrigan, C. D. Stern, *et al.*, “Openmm 7: Rapid development of high performance algorithms for molecular dynamics,” *PLoS computational biology*, vol. 13, no. 7, p. e1005659, 2017.
- [12] M. J. Harvey, G. Giupponi, and G. D. Fabritiis, “Acemd: accelerating biomolecular dynamics in the microsecond time scale,” *Journal of chemical theory and computation*, vol. 5, no. 6, pp. 1632–1639, 2009.
- [13] R. Salomon-Ferrer, A. W. Gotz, D. Poole, S. Le Grand, and R. C. Walker, “Routine microsecond molecular dynamics simulations with amber on gpus. 2. explicit solvent particle mesh ewald,” *Journal of chemical theory and computation*, vol. 9, no. 9, pp. 3878–3888, 2013.
- [14] S. S. Schoenholz and E. D. Cubuk, “Jax, m.d.: End-to-end differentiable, hardware accelerated, molecular dynamics in pure python,” 2019.
- [15] W. Wang, S. Axelrod, and R. Gómez-Bombarelli, “Differentiable molecular simulations for control and learning,” 2020.
- [16] D. L. Mobley, C. C. Bannan, A. Rizzi, C. I. Bayly, J. D. Chodera, V. T. Lim, N. M. Lim, K. A. Beauchamp, D. R. Slochow, M. R. Shirts, *et al.*, “Escaping atom types in force fields using direct chemical perception,” *Journal of chemical theory and computation*, vol. 14, no. 11, pp. 6076–6092, 2018.
- [17] L.-P. Wang, J. Chen, and T. V. Voorhis, “Systematic parametrization of polarizable force fields from quantum chemistry data,” *Journal of Chemical Theory and Computation*, vol. 9, pp. 452–460, Nov. 2012.
- [18] L.-P. Wang, T. J. Martinez, and V. S. Pande, “Building force fields: An automatic, systematic, and reproducible approach,” *The Journal of Physical Chemistry Letters*, vol. 5, pp. 1885–1891, May 2014.
- [19] Y. Qiu, P. S. Nerenberg, T. Head-Gordon, and L.-P. Wang, “Systematic optimization of water models using liquid/vapor surface tension data,” *The Journal of Physical Chemistry B*, vol. 123, pp. 7061–7073, July 2019.
- [20] R. L. Murphy, B. Srinivasan, V. A. Rao, and B. Ribeiro, “Janosy pooling: Learning deep permutation-invariant functions for variable-size inputs,” *CoRR*, vol. abs/1811.01900, 2018.
- [21] Y. Wang, J. Fass, C. D. Stern, K. Luo, and J. Chodera, “Graph nets for partial charge prediction,” *arXiv preprint arXiv:1909.07903*, 2019.
- [22] K. Schütt, P.-J. Kindermans, H. E. S. Felix, S. Chmiela, A. Tkatchenko, and K.-R. Müller, “Schnet: A continuous-filter convolutional neural network for modeling quantum interactions,” in *Advances in neural information processing systems*, pp. 991–1001, 2017.
- [23] K. Yao, J. E. Herr, D. Toth, R. Mckintyre, and J. Parkhill, “The tensormol-0.1 model chemistry: a neural network augmented with long-range physics,” *Chem. Sci.*, vol. 9, pp. 2261–2269, 2018.
- [24] J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman, and D. A. Case, “Development and testing of a general amber force field,” *Journal of Computational Chemistry*, vol. 25, no. 9, pp. 1157–1174, 2004.
- [25] J. Wang, W. Wang, P. A. Kollman, and D. A. Case, “Automatic atom type and bond type perception in molecular mechanical calculations,” *Journal of molecular graphics and modelling*, vol. 25, no. 2, pp. 247–260, 2006.
- [26] K. Vanommeslaeghe, E. P. Raman, and A. D. MacKerell Jr, “Automation of the charmm general force field (cgenff) ii: assignment of bonded parameters and partial atomic charges,” *Journal of chemical information and modeling*, vol. 52, no. 12, pp. 3155–3168, 2012.
- [27] K. Vanommeslaeghe, E. P. Raman, and A. D. MacKerell Jr, “Automation of the charmm general force field (cgenff) ii: assignment of bonded parameters and partial atomic charges,” *Journal of chemical information and modeling*, vol. 52, no. 12, pp. 3155–3168, 2012.
- [28] B. Weisfeiler and A. Leman, “The reduction of a graph to canonical form and the algebgra which appears therein,” 1968.
- [29] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How powerful are graph neural networks?,” *arXiv preprint arXiv:1810.00826*, 2018.
- [30] “An informal amber small molecule force field: parm@frosst,” 2010.

- [31] J. J. Irwin and B. K. Shoichet, "Zinc- a free database of commercially available compounds for virtual screening," *Journal of chemical information and modeling*, vol. 45, no. 1, pp. 177–182, 2005.
- [32] C. C. Bannan and D. Mobley, "ChemPer: An Open Source Tool for Automatically Generating SMIRKS Patterns," 6 2019.
- [33] J. Wagner, M. Thompson, D. Dotson, hyejang, and J. Rodríguez-Guerra, "openforcefield/openforcefields: Version 1.2.1 "Parsley" Update," Sept. 2020.
- [34] D. L. Mobley, C. C. Bannan, A. Rizzi, C. I. Bayly, J. D. Chodera, V. T. Lim, N. M. Lim, K. A. Beauchamp, M. R. Shirts, M. K. Gilson, *et al.*, "Open force field consortium: Escaping atom types using direct chemical perception with smirnoff v0. 1," *BioRxiv*, p. 286542, 2018.
- [35] D. G. Smith, D. Altarawy, L. A. Burns, M. Welborn, L. N. Naden, L. Ward, S. Ellis, B. P. Pritchard, and T. D. Crawford, "The molssi qcarchive project: An open-source platform to compute, organize, and share quantum chemistry data," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, p. e1491, 2020.
- [36] M. J. Hwang, T. P. Stockfisch, and A. T. Hagler, "Derivation of class ii force fields. 2. derivation and characterization of a class ii force field, cff93, for the alkyl functional group and alkane molecules," *Journal of the American Chemical Society*, vol. 116, no. 6, pp. 2515–2525, 1994.
- [37] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.
- [38] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li, and Z. Zhang, "Deep graph library: A graph-centric, highly-performant package for graph neural networks," 2019.
- [39] J. Wagner, D. L. Mobley, C. Bannan, J. Chodera, A. Rizzi, M. Thompson, J. Horton, D. Dotson, J. Rodríguez-Guerra, Camila, C. Bayly, JoshHorton, trevorgokey, N. M. Lim, V. Lim, P. Behara, SimonBoothroyd, S. Sasmal, D. Smith, Lee-Ping, and Y. Zhao, "openforcefield/openforcefield: 0.7.2 Bugfix and minor feature release," Sept. 2020.
- [40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [41] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *CoRR*, vol. abs/1609.02907, 2016.
- [42] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in neural information processing systems*, pp. 1024–1034, 2017.
- [43] "RDKit: Open-source cheminformatics." <http://www.rdkit.org>, 2013. [Online; accessed 11-April-2013].
- [44] J. R. Maple, M.-J. Hwang, T. P. Stockfisch, U. Dinur, M. Waldman, C. S. Ewig, and A. T. Hagler, "Derivation of class ii force fields. i. methodology and quantum force field for the alkyl functional group and alkane molecules," *Journal of Computational Chemistry*, vol. 15, no. 2, pp. 162–182, 1994.
- [45] M. J. Hwang, T. Stockfisch, and A. Hagler, "Derivation of class ii force fields. 2. derivation and characterization of a class ii force field, cff93, for the alkyl functional group and alkane molecules," *Journal of the American Chemical Society*, vol. 116, no. 6, pp. 2515–2525, 1994.
- [46] J. Maple, M.-J. Hwang, T. Stockfisch, and A. Hagler, "Derivation of class ii force fields. iii. characterization of a quantum force field for alkanes," *Israel Journal of Chemistry*, vol. 34, no. 2, pp. 195–231, 1994.
- [47] Z. Peng, C. S. Ewig, M.-J. Hwang, M. Waldman, and A. T. Hagler, "Derivation of class ii force fields. 4. van der waals parameters of alkali metal cations and halide anions," *The Journal of Physical Chemistry A*, vol. 101, no. 39, pp. 7243–7252, 1997.
- [48] J. Maple, M.-J. Hwang, K. J. Jalkanen, T. P. Stockfisch, and A. T. Hagler, "Derivation of class ii force fields: V. quantum force field for amides, peptides, and related compounds," *Journal of computational chemistry*, vol. 19, no. 4, pp. 430–458, 1998.
- [49] P. Dauber-Osguthorpe and A. T. Hagler, "Biomolecular force fields: where have we been, where are we now, where do we need to go and how do we get there?," *Journal of computer-aided molecular design*, vol. 33, no. 2, pp. 133–203, 2019.

- [50] A. T. Hagler, "Force field development phase ii: Relaxation of physics-based criteria. . . or inclusion of more rigorous physics into the representation of molecular energetics," *Journal of computer-aided molecular design*, vol. 33, no. 2, pp. 205–264, 2019.
- [51] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, *et al.*, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint arXiv:1806.01261*, 2018.
- [52] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.
- [53] J. Du, S. Zhang, G. Wu, J. M. Moura, and S. Kar, "Topology adaptive graph convolutional networks," *arXiv preprint arXiv:1710.10370*, 2017.
- [54] F. Wu, T. Zhang, A. H. d. Souza Jr, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," *arXiv preprint arXiv:1902.07153*, 2019.
- [55] A. S. Christensen and O. A. von Lilienfeld, "On the role of gradients for machine learning of molecular energies and forces," 2020.
- [56] S. Chmiela, H. E. Sauceda, I. Poltavsky, K.-R. Müller, and A. Tkatchenko, "sgdml: Constructing accurate and data efficient molecular force fields using machine learning," *Computer Physics Communications*, vol. 240, pp. 38–45, 2019.
- [57] H. E. Sauceda, S. Chmiela, I. Poltavsky, K.-R. Müller, and A. Tkatchenko, "Molecular force fields with gradient-domain machine learning: Construction and application to dynamics of small molecules with coupled cluster forces," *The Journal of Chemical Physics*, vol. 150, no. 11, p. 114102, 2019.
- [58] H. E. Sauceda, M. Gastegger, S. Chmiela, K.-R. Müller, and A. Tkatchenko, "Molecular force fields with gradient-domain machine learning (gdml): Comparison and synergies with classical force fields," 2020.

6 Code Availability

The Python code used to produce the results discussed in this paper is distributed open source under MIT license [<https://github.com/choderalab/espaloma>]. Core dependencies include PyTorch [37], Deep Graph Library [38], the Open Force Field Toolkit [34, 39], and OpenMM [11].

7 Experimental Details

All datasets, including the ZINC that is distributed with parm@Frosst [30], AlkEthOH [32], and QCArchive OpenForceField Optimization Set 1 [35], are split randomly across *chemical species* rather than *snapshots* into training:test=80:20. All models are trained with 5000 epochs with Adam optimizer [40]. Hyperparameters, namely choices of graph net layer architectures, activation functions, learning rates, and per-layer units, are optimized with a grid search using training sets. All experiments are done with three layers of graph convolutional neural networks with architectures introduced in 12.2 without dropout. GCN [41] with 32 units and leaky ReLU activation function is used for experiments in 3.1. GraphSAGE [42] with 128 units and sigmoid activation function is used for experiments in 1 and 3. The input features of the atoms included the one-hot encoded element number, as well as the hybridization, aromaticity, and formal charge thereof, were assigned using Open Force Field toolkit [39] and RDKit [43]. High-temperature MD simulations described in Section 1 were initialized using RDKit’s default conformer generator followed by OpenMM’s energy minimizer, with random initial velocities at the target temperature. Trajectories were simulated using OpenMM [11]’s default LangevinIntegrator, with a temperature of 500 K, collision rate of 1/picosecond, and a timestep of 1 fs. Samples were saved every 10 steps. Reported metrics: R^2 : the coefficient of determination, RMSE: root mean square error, MAPE: mean absolute percentage error; note that the MAPE results we report is not multiplied by 100, and therefore denotes the fractional error. The annotated 95% confidence intervals are calculated by bootstrapping the test set 1000 times.

8 Funding

YW acknowledges support from NSF CHI-1904822 and the Sloan Kettering Institute. JF acknowledges support from NSF CHE-1738979 and the Sloan Kettering Institute. JDC acknowledges support from NIH grant P30 CA008748, NIH grant R01 GM121505, NIH grant R01 GM132386, NSF CHI-1904822, and the Sloan Kettering Institute.

9 Disclosures

YW is among the co-founders and equity holders of Uli, Inc. and Uli (Shenzhen) Technology Co. Ltd.

JDC is a current member of the Scientific Advisory Board of OpenEye Scientific Software. The Chodera laboratory receives or has received funding from multiple sources, including the National Institutes of Health, the National Science Foundation, the Parker Institute for Cancer Immunotherapy, Relay Therapeutics, Entasis Therapeutics, Silicon Therapeutics, EMD Serono (Merck KGaA), AstraZeneca, Vir Biotechnology, XtalPi, the Molecular Sciences Software Institute, the Starr Cancer Consortium, the Open Force Field Consortium, Cycle for Survival, a Louis V. Gerstner Young Investigator Award, and the Sloan Kettering Institute. A complete funding history for the Chodera lab can be found at <http://choderalab.org/funding>.

10 Disclaimers

The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

11 Author Contributions

Conceptualization: JF, YW, JDC; Data Curation: YW; Formal Analysis: YW; Funding Acquisition: JDC; Investigation: YW, JF; Methodology: YW, JF; Project Administration: JDC; Resources: JDC; Software: YW; Supervision: JDC; Visualization: YW; Writing – Original Draft: YW; Writing – Review & Editing: YW, JDC, JF.

12 Appendix

12.1 Class I molecular mechanics force fields

In a class I molecular mechanics force field [44, 45, 46, 47, 48, 49, 50], Φ_{FF} , the valence (bonded) portion of potential energy of a single molecule \mathcal{G} in vacuum with conformation $\mathbf{x} \in \mathbb{R}^{|\mathcal{G}|*3}$ can be factorized as bond, angle, and torsion energy,

$$U_{\Phi_{\text{FF}}}(\mathbf{x}, \mathcal{G}) = \sum_{r \in \mathcal{G}_{\text{bond}}} U_{\Phi_{\text{FF}}, \text{bond}}(r(\mathbf{x})) + \sum_{\theta \in \mathcal{G}_{\text{angle}}} U_{\Phi_{\text{FF}}, \text{angle}}(\theta(\mathbf{x})) + \sum_{\phi \in \mathcal{G}_{\text{torsion}}} U_{\Phi_{\text{FF}}, \text{torsion}}(\phi(\mathbf{x})), \quad (6)$$

where the subscripts $\mathcal{G}_{\text{bond}}$, $\mathcal{G}_{\text{angle}}$, $\mathcal{G}_{\text{torsion}}$ denote the duplet, triplet, and quadruplet subsets of the groups of vertices in \mathcal{G} . If we use $\mathcal{N}(\cdot)$ to denote the operation to find the set of neighbors,

$$\mathcal{G}_{\text{bond}} = \{r_{ij} = (v_i, v_j), v_i \neq v_j, v_i \in \mathcal{N}(v_j)\}; \quad (7)$$

$$\mathcal{G}_{\text{angle}} = \{\theta_{ijk} = (v_i, v_j, v_k), v_i \neq v_j \neq v_k, v_i \in \mathcal{N}v_j, v_j \in \mathcal{N}v_k\}; \quad (8)$$

$$\mathcal{G}_{\text{torsion}} = \{\phi_{ijkl} = (v_i, v_j, v_k, v_l), v_i \neq v_j \neq v_k \neq v_l, v_i \in \mathcal{N}(v_j), v_j \in \mathcal{N}(v_k), v_k \in \mathcal{N}(v_l)\}. \quad (9)$$

The composing energy functions usually take harmonic or periodic forms w.r.t. the lengths of bonds and the values of angles and torsions. With tabulated collections of parameters

$\{\mathbf{k}_{\text{bond}}, \mathbf{b}_{\text{bond}}, \mathbf{k}_{\text{angle}}, \mathbf{b}_{\text{angle}}, \mathbf{k}_{\text{torsion}}, \phi_{0,\text{torsion}}\}$ we have:

$$U_{\Phi_{\text{FF}}, \text{bond}}(r_{ij}(\mathbf{x})) = \frac{1}{2}k_{\text{bond},r}(r_{ij}(\mathbf{x}) - b_{\text{bond},r})^2; r_{ij}(\mathbf{x}) = \|\mathbf{x}_i - \mathbf{x}_j\|; \quad (10)$$

$$U_{\Phi_{\text{FF}}, \text{angle}}(\theta_{ijk}(\mathbf{x})) = \frac{1}{2}k_{\text{angle},\theta}(\theta_{ijk}(\mathbf{x}) - b_{\text{angle},\theta})^2; \theta_{ijk}(\mathbf{x}) = \langle \mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k \rangle; \quad (11)$$

$$U_{\Phi_{\text{FF}}, \text{torsion}}(\theta_{ijkl}(\mathbf{x})) = \sum_{n=1}^{n_{\text{max}}} k_{\text{torsion},\phi,n}(1 + \cos(n\phi - \phi_{0,n})); \phi_{ijkl}(\mathbf{x}) = \langle \mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k, \mathbf{x}_l \rangle, \quad (12)$$

where the bracket denote the angle between three points or the dihedral angle between the two planes formed by the two sets three consecutive points among the four points on \mathbb{R}^3 . Note that, in our implementation, we do not include improper torsion terms as they are sparsely assigned in legacy force fields.

12.2 Graph Nets

In the context of molecular machine learning, molecules are modelled as undirected graphs of bonded atoms, where each atom and bond can carry attributes reflecting their chemical nature from which complex chemical features can be learned. If we write this as a tuple of three sets,

$$g = \{\mathcal{V}, \mathcal{E}, \mathcal{U}\} \quad (13)$$

Here, \mathcal{V} is the set of the vertices (or nodes) (atoms), \mathcal{E} the set of edges (bonds), and $\mathcal{U} = \{\mathbf{u}\}$ the universal (global) attribute.

A set of functions (with learnable parameters) govern the three stages used in both training and inference of a graph net: *initialization*, *propagation*, and *readout*. The most general description of the message-passing procedure in the propagation stage could be found in [51], where node, edge, and global attributes \mathbf{v} , \mathbf{e} , \mathbf{u} are updated according to:

$$\mathbf{e}_k^{(t+1)} = \phi^e(\mathbf{e}_k^{(t)}, \sum_{i \in \mathcal{N}_k^e} \mathbf{v}_i, \mathbf{u}^{(t)}), \quad \text{edge update} \quad (14)$$

$$\bar{\mathbf{e}}_i^{(t+1)} = \rho^{e \rightarrow v}(E_i^{(t+1)}), \quad \text{edge-to-node aggregate} \quad (15)$$

$$\mathbf{v}_i^{(t+1)} = \phi^v(\bar{\mathbf{e}}_i^{(t+1)}, \mathbf{v}_i^{(t)}, \mathbf{u}^{(t)}), \quad \text{node update} \quad (16)$$

$$\bar{\mathbf{e}}^{(t+1)} = \rho^{e \rightarrow u}(E^{(t+1)}), \quad \text{edge-to-global aggregate} \quad (17)$$

$$\bar{\mathbf{u}}^{(t+1)} = \rho^{v \rightarrow u}(V^{(t)}), \quad \text{node-to-global aggregate} \quad (18)$$

$$\mathbf{u}^{(t+1)} = \phi^u(\bar{\mathbf{e}}^{(t+1)}, \bar{\mathbf{v}}^{(t+1)}, \mathbf{u}^{(t)}), \quad \text{global update} \quad (19)$$

where $E_i = \{\mathbf{e}_k, k \in \mathcal{N}_i^v\}$ is the set of attributes of edges connected to a specific node, $E_i = \{\mathbf{e}_k, k \in 1, 2, \dots, N^e\}$ is the set of attributes of all edges, V is the set of attributes of all nodes, and \mathcal{N}^v and \mathcal{N}^e denote the set of indices of entities connected to a certain node or a certain edge, respectively. ϕ^e , ϕ^v , and ϕ^u are update functions that take the *environment* of the an entity as input and update the attribute of the entity, which could be stateful or not; $\rho^{e \rightarrow v}$, $\rho^{e \rightarrow u}$, and $\rho^{v \rightarrow u}$ are aggregate functions that aggregate the attributes of multiple entities into an *aggregated* attribute which shares the same dimension with each entity. Note that it is common that the edges do not hold attribute but only pass message onto neighboring nodes. For all models we survey here, edge-to-global update does not apply and global attribute does not present until the readout stage, when a sum function is applied to form the global representation ($\mathbf{u} = \sum V$). Under this set of grammar, we review the message-passing rules in Table 2.

13 Training losses

As mentioned in section 2.2, we can train the model using an arbitrary differentiable loss. Here, we describe the loss functions used in the reported experiments.

Model	Edge update ϕ^e	Edge aggregate $\rho^{e \rightarrow v}$	Node update ϕ^v
GCN	Identity	Mean	NN
EdgeConv	$\text{ReLU}(W_0(\mathbf{v}_i - \mathbf{v}_j) + W_1\mathbf{v}_i)$	Max	Identity
GraphSAGE	Identity	Mean*	Normalize($\text{NN}([\mathbf{v} : \mathbf{e}])$)
GIN	Identity	Sum*	$\text{NN}((1 + \epsilon)\mathbf{v} + \mathbf{e})$

Table 2: Summary of representative graph nets architectures by edge update, edge aggregate, and node update types. Models analyzed here include: GCN [41], EdgeConv [52], GraphSAGE [42], and GIN [29]. Other architectures evaluated—TAGCN [53] and SGC [54]—involve multi-step propagation, which could be expressed as a combination of these updates and aggregates.

*: Multiple aggregation functions studied in the referenced publication.

13.1 Classification loss

In Section 3.1, we used the cross-entropy loss, as implemented in PyTorch³, to reproduce GAFF 1.81 atom types.

13.2 Training by potential energies

Given a training set of molecules $\{\mathcal{G}_i\}$ with corresponding snapshots $\{\mathbf{x}_i \in \mathbf{R}^{(|\mathcal{G}_i| \times 3)}\}$ and potential energies $\{U_{i,\text{ref}}\}$ the model parameters W_{NN} can be optimized to maximize likelihood of the reference energy under Gaussian noise model (assuming fixed noise) where the mean of the predictive distribution is given by the composed force field $\Phi_{\text{FF},\Theta}$. When fitting to QM targets, since there is an offset in QM energies that cannot be accounted for using MM functional forms, we subtract per-molecule mean for both predicted and reference energies. Note that more sophisticated strategies that penalize higher-energy (lower-probability) snapshots can also be employed, depending on the application.

13.3 Training by forces:

To exploit the rich information in the forces [55, 56, 57, 58] $\frac{\partial U}{\partial \mathbf{x}}$, which can be computed using auto-differentiating packages, one can jointly optimize the likelihood of the reference energies and forces (which are assumed to be independent).

³<https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html#torch.nn.CrossEntropyLoss>

