HHS Public Access

OpenMM 8: Molecular Dynamics Simulation with Machine Learning Potentials

Peter Eastman[1,*], **Raimondas Galvelis**[2,3], **Raúl P. Peláez**[3], **Charlles R. A. Abreu**[4,5], **Stephen E. Farr**[6], **Emilio Gallicchio**[7,8], **Anton Gorenko**[9], **Michael M. Henry**[10], **Frank Hu**[1], **Jing Huang**[11], **Andreas Krämer**[12], **Julien Michel**[6], **Joshua A. Mitchell**[13], **Vijay S. Pande**[14,15], **João PGLM Rodrigues**[15], **Jaime Rodriguez-Guerra**[16], **Andrew C. Simmonett**[17], **Sukrit Singh**[10], **Jason Swails**[18], **Philip Turner**[19], **Yuanqing Wang**[20], **Ivy Zhang**[10,21], **John D. Chodera**[10], **Gianni De Fabritiis**[2,3,22], **Thomas E. Markland**[1]

[1]Department of Chemistry, Stanford University, Stanford, CA 94305, USA

[2]Acellera Labs, C Dr Trueta 183, 08005, Barcelona, Spain

[3]Computational Science Laboratory, Universitat Pompeu Fabra, Barcelona Biomedical Research Park (PRBB), C Dr. Aiguader 88, 08003, Barcelona, Spain

[4]Chemical Engineering Department, School of Chemistry, Federal University of Rio de Janeiro, Rio de Janeiro 68542, Brazil

[5]Redesign Science Inc., 180 Varick St., New York, NY 10014, USA

[6]EaStCHEM School of Chemistry, University of Edinburgh, EH9 3FJ, United Kingdom

[7]Department of Chemistry and Biochemistry, Brooklyn College of the City University of New York, NY, USA

[8]Ph.D. Program in Chemistry and Ph.D. Program in Biochemistry, The Graduate Center of the City University of New York, New York, NY, USA

[9]Stream HPC, Koningin Wilhelminaplein 1 - 40601, 1062 HG Amsterdam, Netherlands

[10]Computational and Systems Biology Program, Sloan Kettering Institute, Memorial Sloan Kettering Cancer Center, New York NY 10065, USA

[11]Key Laboratory of Structural Biology of Zhejiang Province, School of Life Sciences, Westlake University, 18 Shilongshan Road, Hangzhou 310024, Zhejiang, China

[12]Department of Mathematics and Computer Science, Freie Universität Berlin, Arnimallee 12, 14195 Berlin, Germany

[13]The Open Force Field Initiative, Open Molecular Software Foundation, Davis, CA 95616, USA

[14]Andreessen Horowitz, 2865 Sand Hill Rd, Menlo Park, CA 94025, USA

[15]Department of Structural Biology, Stanford University, Stanford, CA 94305, USA

---

*Corresponding author (peastman@stanford.edu).

[16]Charité Universitätsmedizin Berlin In silico Toxicology and Structural Bioinformatics, Virchowweg 6, 10117 Berlin, Germany

[17]Laboratory of Computational Biology, National Heart, Lung and Blood Institute, National Institutes of Health, Bethesda, MD 20892, USA

[18]Entos Inc., 9310 Athena Circle, La Jolla, CA 92037, USA

[19]College of Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA

[20]Simons Center for Computational Physical Chemistry and Center for Data Science, New York University, 24 Waverly Place, New York, NY 10004, USA

[21]Tri-Institutional PhD Program in Computational Biology and Medicine, Weill Cornell Medical College, Cornell University, New York, NY 10065, USA

[22]ICREA, Passeig Lluis Companys 23, 08010, Barcelona, Spain

Abstract

Machine learning plays an important and growing role in molecular simulation. The newest version of the OpenMM molecular dynamics toolkit introduces new features to support the use of machine learning potentials. Arbitrary PyTorch models can be added to a simulation and used to compute forces and energy. A higher-level interface allows users to easily model their molecules of interest with general purpose, pretrained potential functions. A collection of optimized CUDA kernels and custom PyTorch operations greatly improves the speed of simulations. We demonstrate these features on simulations of cyclin-dependent kinase 8 (CDK8) and the green fluorescent protein (GFP) chromophore in water. Taken together, these features make it practical to use machine learning to improve the accuracy of simulations at only a modest increase in cost.

Introduction

In recent years, much work in the field of molecular simulation has focused on ways to produce more accurate results at lower cost. This includes newer force fields that use better functional forms or better parametrization to improve accuracy without significantly increasing cost. It also includes semi-empirical and machine learning methods that try to approach the accuracy of high-level quantum chemistry (QC) methods at a cost that is intermediate between QC and classical force fields. Another important aspect is the use of newer sampling methods to reduce the amount of simulation needed to obtain converged results for thermodynamic quantities.

In this paper, we describe the newly released version 8 of OpenMM, a popular package for molecular simulation that provides excellent performance and high flexibility.[1] It contains new features to better support these methods, with a particular emphasis on machine learning. They make it a powerful tool both for methodological research and for production simulations.

Machine learning potentials (MLPs) have increasingly emerged as a major area of research within the field of molecular simulation. They offer a powerful intermediate between accurate but slow quantum chemistry methods on the one hand, and fast but less accurate force fields on the other. Unlike force fields, which use simple functions chosen by the designer to compute interactions, MLPs use highly flexible models that can learn very complicated functions, most often implemented as neural networks. The models are trained on molecular energies, forces, and other properties computed with a high-level QC method. The trained models can approach the accuracy of the method used to generate the training data, while being orders of magnitude faster.[2]

The past few years have seen an explosion of new MLP architectures designed to improve their accuracy, speed, transferability, and data efficiency. Many of these architectures try to mirror the symmetries obeyed by the physical system, particularly equivariance under translations and rotations.[3–7] Others try to improve transferability by directly incorporating a limited amount of physics into the model, such as explicit terms for Coulomb interactions, dispersion, and nuclear repulsion.[8–11] Still others take this approach even further, beginning with a semi-empirical quantum chemistry method and training a machine learning model to correct for its flaws and improve its accuracy.[12–14] These approaches provide different tradeoffs between speed, accuracy, and range of applicability, with new architectures continuing to be introduced.

Another important area of research is ways of combining MLPs with classical force fields. Although they are much faster than high-level quantum chemistry algorithms, they are still orders of magnitude slower than most force fields. This often makes them impractical for simulating macromolecules or other large systems. Just as QM/MM simulations use quantum mechanics for only a small piece of a system while simulating the rest with classical mechanics, ML/MM simulations can simulate part with machine learning and the rest with a classical force field. This can lead to large improvements in accuracy with only a modest increase in computational cost.[15,16]

We have recently released version 8 of the OpenMM simulation package, with a focus on supporting machine learning potentials. OpenMM is a widely used piece of software for molecular simulation, especially biological macromolecules, and one of the fastest growing engines for atomistic molecular simulations[17]. Its architecture and features provide a unique combination of speed, flexibility, and extensibility. Some of the more notable aspects include its excellent performance when running on GPUs; the ability to implement entirely new algorithms and simulation protocols through Python scripting; and the ability to define entirely new functional forms for interactions, which are transparently compiled to machine code and executed with no loss of performance on the GPU or CPU.

OpenMM 8 extends this flexibility to MLPs. The new features described below aim to achieve two goals. First, they provide a powerful environment for researchers developing new MLPs or new simulation protocols. Arbitrary PyTorch models can be easily added to a simulation and used to compute forces and energies. Second, they bridge the gap to production simulations, allowing end users to select from a set of validated general-purpose

MLPs and run simulations with them. With these new features, setting up and running a ML or ML/MM simulation is no more difficult than using a conventional force field.

Methods

PyTorch Model Support

The foundation of MLP support is the OpenMM-Torch module. It provides the TorchForce class, which allows arbitrary PyTorch[18] models to be embedded in a simulation and used to calculate forces and energies. PyTorch has emerged as the standard machine learning framework within the simulation community, and nearly all MLP models published in recent years have been implemented with it. By supporting it, we allow existing models to easily interface with OpenMM. It also provides an ideal environment for methodological research, since model developers can continue using the same framework they are already accustomed to. Interfaces to TensorFlow and TensorRT are also available, but because the community has largely standardized on PyTorch, they are not discussed further here.

To create a TorchForce, one simply provides a model that takes atomic positions as input and computes potential energy. Forces can be automatically calculated through backpropagation, or alternatively, the model can return them explicitly. All features of PyTorch are supported. The only restriction on the model is that it must be possible to compile it to TorchScript. The following code is a minimalist example that illustrates how to apply a potential function computed with PyTorch, in this case a harmonic potential attracting every particle to the origin.

```
Class ForceModule(torch.nn.Module):
    def forward(self, positions):
        return torch.sum(positions**2)
module = torch.jit.script(ForceModule())
force = TorchForce(module)
```

OpenMM-ML

The next level in the stack is the OpenMM-ML module, which provides a convenient mechanism for running simulations with pretrained potentials. It makes MLPs as simple to use as conventional force fields. For example, given a Topology object (a description of the atoms and chemical bonds making up a system to be simulated), the following code prepares a simulation of it with the ANI-2x[19] potential function.

```
potential = MLPotential('ani2x')
system = potential.createSystem(topology)
```

Mixed simulations, in which a system is modeled partly with a conventional force field and partly with a MLP, are just as easy to prepare. One simply provides a version that is modeled entirely with the force field, and a list of which atoms make up the subset to model with

ML. All details of setting up the mixed system, including handling the interaction of the two regions through mechanical embedding, are handled automatically.

```
forcefield = ForceField('amber14-all.xml', 'amber14/tip3pfb.xml')
mm_system = forcefield.createSystem(topology)
potential = MLPotential('ani2x')
ml_system = potential.createMixedSystem(topology, mm_system, ml_atoms)
```

At present, the only supported MLPs are ANI-1ccx[20] and ANI-2x[19]. These potentials are fast and have reasonably good accuracy, but their range of applicability is limited. They support only a limited set of elements (seven for ANI-2x, four for ANI-1ccx), and do not support charged molecules. We intend that the list of potentials will grow with time as new pretrained MLPs are developed and made available to the community.

To facilitate the development of new models, we have created the SPICE (Small-Molecule/ Protein Interaction Chemical Energies) dataset[21], a large collection of molecular forces and energies calculated at the highly accurate ωB97M-D3BJ/def2-TZVPPD[22] level of theory. It incorporates a wide selection of molecules relevant to simulating drug-like small molecules interacting with proteins. It includes 15 elements, both charged and neutral molecules, both low and high energy conformations, and a wide assortment of covalent and non-covalent interactions. We hope that with time it will lead to the creation of many new pretrained MLPs that can be made available for use by the simulation community.

Optimization Through NNPOps

For MLPs to be useful, they must be fast. For this reason, we created the NNPOps module with the goal of accelerating machine learning models. It contains optimized CUDA kernels and PyTorch code for bottleneck operations in important MLPs. Examples include computing the atom-centered symmetry functions used in the ANI family of potentials, building neighbor lists, and computing electrostatic interactions with the Particle Mesh Ewald method. Benchmarks demonstrating the speedup from NNPOps are in the next section.

Other Features

While OpenMM 8 includes important new features for machine learning, numerous other new features have been added in the years since version 7 was released in 2016. We highlight here a few of the more significant ones.

Enhanced Sampling Methods—Enhanced sampling algorithms allow a simulation to explore conformation space more efficiently and produce converged results in less time. A number of methods are available for use with OpenMM.

Metadynamics[23] is supported through two different mechanisms. One is based on the popular PLUMED library[24], which offers a wide selection of useful collective variables (CVs) along which to accelerate exploration. The other is a native implementation that uses a novel mechanism to define CVs. Any energy term that can be defined with OpenMM's

force classes can be used as a CV. Given OpenMM's flexible custom forces supporting arbitrary user-defined functional forms, this is a very powerful mechanism.

To illustrate the flexibility of this approach, we present a few examples of how common CVs can be defined. One common choice is a weighted sum of distances between particular pairs of atoms. This is easily defined with OpenMM's CustomBondForce.

```
cv = CustomBondForce('weight*r')
cv.addPerBondParameter('weight')
for atom1, atom2, weight in pairs:
    cv.addBond(atom1, atom2, [weight])
```

In other cases a weighted sum of dihedral angles could be more appropriate. This is implemented just as easily with CustomTorsionForce.

```
cv = CustomTorsionForce('weight*theta')
cv.addPerTorsionParameter('weight')
for atom1, atom2, atom3, atom4, weight in dihedrals:
    cv.addTorsion(atom1, atom2, atom3, atom4, [weight])
```

More complicated functions of distances, angles, and dihedrals can be defined just as easily, as can terms that depend on the centroids of groups of atoms, root mean squared deviations (RMSD) between sets of atoms, and many other functions. One can even use the TorchForce class described above to define CVs through PyTorch code.

For situations when specific CVs are not known in advance, an implementation of simulated tempering[25] has been added. It accelerates transitions across barriers by allowing a simulation to temporarily increase its temperature. It joins other methods that were already available in OpenMM and its associated packages, including aMD[26], replica exchange[27], and Replica Exchange with Solute Tempering (REST)[28].

Another new algorithm is the Alchemical Transfer Method[29] (ATM), which provides a convenient and efficient method of computing free energy differences. Interaction forces are computed before and after performing a coordinate transformation, such as translating a molecule between the binding pocket and bulk solvent. The simulation evolves based on a weighted average of the two. This permits efficient sampling of alchemical pathways used to compute absolute or relative free energies. Unlike many other alchemical methods, it does not require soft-core interactions and can be applied to arbitrary potential functions, including ones defined by MLPs.

New Force Fields—Classical force fields continue to advance, improving both in accuracy and range of applicability. Many new force fields are now supported. This includes newer versions of the Amber[30] and CHARMM[31] point charge force fields and the polarizable AMOEBA[32] and CHARMM Drude[33] force fields. The GLYCAM[34] force

field has been added for simulating carbohydrates. For simulating arbitrary small organic molecules, the OpenFF[35] and GAFF[36] generic force fields are now available.

New Integrators—OpenMM has added support for the LF-Middle integration algorithm[37,38]. This is an improved discretization of the Langevin equation that significantly reduces error compared to previously used discretizations. In many cases, it allows doubling the step size with no loss in accuracy, leading to a huge increase in performance.

Other new integrators have also been added, including a Nosé-Hoover integrator[39], a dual-temperature version of the Nosé-Hoover integrator for simulating Drude particles[40], and a multiple time step Langevin integrator. In addition, many features have been added to the CustomIntegrator class, which allows users to define entirely new integration algorithms. These features greatly increase the range of methods that can be implemented with it.

Improved Hardware Support—OpenMM has grown in its ability to take advantage of a range of hardware. ARM and PowerPC processors are now supported. A new backend based on the HIP framework provides greatly improved performance on AMD GPUs (see Table 1). The ability to efficiently parallelize a simulation across multiple GPUs has improved (see Table 2). The speed of running simulations on CPUs has also greatly improved.

Results and Discussion

Performance of MLPs

To demonstrate the effect of the optimizations in NNPOps, we simulated cyclin-dependent kinase 8 (CDK8) bound to a 53 atom inhibitor (lig_30 from the protein-ligand-benchmark set[41], shown in Figure 1). CDK8 has 632 residues and 10,456 atoms. We modeled it with the Amber14 force field and solvated it with 41,194 TIP3P-FB water molecules, giving a total of 134,099 atoms.

We compared the simulation speed when modeling the inhibitor either with the classical OpenFF 2.0.0 force field[35] or with the ANI-2x machine learning potential. Interactions between the inhibitor and the rest of the system were computed with the Amber14 force field in all cases. We further compared the speed of ANI-2x using either the standard implementation from the TorchANI library or the optimized version from NNPOps.

ANI-2x is an ensemble of eight models that were each trained independently. The average of the models is slightly more accurate than any one of the models on its own. When speed is important, it is often acceptable to use just one of the eight models. This produces a large improvement in speed in exchange for a small loss in accuracy. We benchmarked this option as well.

All of the simulations described above were run with a Langevin integrator and a 2 fs time step. In classical simulations, it is common to constrain the lengths of bonds involving hydrogen and to increase the mass of hydrogen atoms through mass repartitioning. This combination allows integration to be stable with a 4 fs time step. In principle, the same optimization could be used for the MLP, but it would require careful selection of the

constrained bond lengths and a thorough validation of the results. Since this optimization is not yet common practice in ML/MM simulations, we chose to omit constraints within the ligand and limit the time step to 2 fs. For the classical force field, we measured the speed with both 2 fs and 4 fs time steps.

Each simulation was integrated for 10,000 steps, and the speed in ns/day was calculated based on the elapsed time. These very short simulations are intended only to benchmark speed. Our goal in this section is not to validate the accuracy of the ANI-2x potential, which has been studied elsewhere[19], only to measure the effect of our optimizations.

All simulations were run on a NVIDIA RTX 4080 GPU. The results are shown in Table 3.

NNPOps is faster than TorchANI by a factor of 5.7, enormously increasing the practicality of using the potential for real simulations. Using a single model instead of the full ensemble improves the speed by another factor of 1.9. Combining these two optimizations, modeling the ligand with ANI-2x is only 2.7 times slower than a fully classical simulation with the same step size. This is a very reasonable cost, given that it has been shown to significantly improve the accuracy in calculated binding free energies. Using a larger step size in the classical simulation increases the ratio to 5.1 times slower. In principle, the larger step size could also be used for ML/MM simulations, but one would need to evaluate the effects of constraining bond lengths on the results.

Viewed another way, these results illustrate how expensive current MLPs still are compared to classical force fields; simulating 53 atoms with ANI-2x is more expensive than simulating the other 134,046 atoms with Amber14. On the other hand, it is orders of magnitude faster than conventional QC methods with similar accuracy. Simulating core pieces of a system with MLPs is now entirely practical, and can yield large improvements to accuracy.

Example: MLP for Anionic Green Fluorescent Protein Chromophore in Water

As a demonstration of the TorchForce class and OpenMM's MLP integration capabilities, we trained an MLP to simulate anionic GFP chromophore (*p*-hydroxybenzylidene-2,3-dimethylimidazolinone, HBDI-)[42] in water. GFP is a widely used genetically-encoded fluorescent tag in biological systems that features a covalently attached chromophore formed by an autocatalytic reaction following folding[43]. The chromophore's fluorescent properties are strongly influenced by its environment. For that reason, it is useful to study its behavior in a variety of environments, including water and other solvents. The system consists of the anionic GFP chromophore with 167 solvating waters for a total of 528 atoms in a periodic orthorhombic cell. A picture of the GFP chromophore and the surrounding water molecules can be seen in Figure 2.

The training data was obtained from *ab initio* molecular dynamics (AIMD) trajectories of the GFP chromophore in water (75008 configurations from 10 separate trajectories) supplemented with 7500 configurations obtained from *ab initio* path integral molecular dynamics (AI-PIMD) trajectories, which incorporate nuclear quantum effects in the sampling. In these trajectories, obtained from our previous work[44], forces were evaluated with CP2K[45] using DFT with the revPBE exchange-correlation functional[46,47] and D3

dispersion corrections[48] added. All trajectories used multiple time step (MTS) integration of the r-RESPA form[49], with an outer time step of 2.0 fs and an inner time step of 0.5 fs. The MTS reference forces were evaluated at a SCC-DFTB3 level of theory[50]. The configurations used for training were taken every 2.0 fs and correspond to 15 ps of AI-PIMD trajectory and ~150 ps of classical AIMD trajectories.

The machine learning architecture that we employ is an equivariant transformer neural network using the TorchMD-NET package[4]. The training data was partitioned randomly into an 80-10-10 split for the training, validation, and test sets respectively, and the model was trained for 400 epochs to all the forces in each configuration. Our model used 2 attention heads, 1 layer, 32 radial basis functions, and an embedding dimension of 64. This resulted in a test set RMSE on forces of 45.7 meV/A. Learning curves for the model are shown in Figure 3.

The trained model was compiled into a TorchScript module and used the TorchForce class to run dynamics in OpenMM 8.0. Using the LangevinMiddleIntegrator, we ran NVT dynamics with a temperature of 300 K, 0.5 fs time step, and a 1.0 ps^{-1} friction coefficient. The final trajectory used for analysis was 1 ns in length, with frames saved every 2.0 fs. Of this trajectory, we discarded the first 15 ps for equilibration.

For comparing our MLP-generated trajectory to the AIMD trajectories, we analyzed the oxygen-oxygen radial distribution functions (RDFs) around the two oxygen atoms on the GFP chromophore as well as the distribution of the ϕ_I and ϕ_P dihedral angles as defined in Figure 2. We chose these two observables because (1) the two oxygen atoms on the chromophore are those that interact the most strongly with the solvent water molecules; and (2) the dihedrals ϕ_I and ϕ_P are the angles that undergo torsional rotation upon photoexcitation of the chromophore[51].

Figure 4 shows the comparison of the RDFs obtained using the MLP and the AIMD where the error bars on the AIMD RDFs are +/− 1 standard deviation across the 10 AIMD trajectories. The MLP and AIMD are in good agreement and within the error bars of the AIMD trajectories. The distribution of dihedral angles ϕ_I and ϕ_P can be seen in Figure 5 and are in similarly good agreement within the error bars.

The advantages of the MLP are evident from the timings. On two AMD EPYC 7763 64 core processors (128 cores total), the AIMD calculation of the force takes 13.2 seconds, whereas the MLP evaluated on a Nvidia RTX 4080 GPU takes only 8.7 ms for each force evaluation, a speed-up of 1517 times.

Conclusions

OpenMM 8 introduces new features to support the use of machine learning potentials in molecular simulations. These features are best viewed as a foundation on which to build future tools. MLPs are still an immature field; few pretrained, general-purpose potentials exist, and those that do exist are limited in their range of applicability. More often, users must train their own special purpose potentials, as we demonstrate here for the GFP chromophore. The tools in OpenMM provide a powerful environment for researchers

developing new pretrained potentials. Those future potentials can then be made easily available to users who wish to run simulations with them.

The tools for running hybrid ML/MM simulations should likewise be viewed as a starting point for future work. For example, an important feature will be to allow the ML and MM regions to interact through electrostatic embedding, rather than the simpler but less accurate mechanical embedding. This will require ML models that can predict atomic partial charges as well as energy.

With time we hope to provide a diverse set of potentials to meet many different needs. To an even greater extent than classical force fields, MLPs can offer a huge range of tradeoffs between speed, accuracy, and range of applicability. Developing a corresponding range of potentials is an important project for the community.

Acknowledgements

Disclosures

JDC is a current member of the Scientific Advisory Board of OpenEye Scientific Software, Redesign Science, Ventus Therapeutics, and Interline Therapeutics, and has equity interests in Interline Therapeutics. 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, Bayer, XtalPi, Interline Therapeutics, 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.

JM is a member of the Scientific Advisory Board of Cresset.

VSP currently sits on the boards of Apeel Sciences, BioAge, Devoted Health, Freenome, Genesis Therapeutics, Insitro, Inceptive Therapeutics, Nautilus Biotechnology, Nobell, Omada Health, Q.bio, SciFi Foods, and Scribe Therapeutics, and is on the Scientific Advisory Board of Schrodinger.

YW has limited financial interests in Flagship Pioneering, Inc. and its subsidiaries.

Availability

OpenMM is available from https://openmm.org. Development is conducted through the Github community at https://github.com/openmm/openmm.

References

(1). Eastman P; Swails J; Chodera JD; McGibbon RT; Zhao Y; Beauchamp KA; Wang L-P; Simmonett AC; Harrigan MP; Stern CD; et al. OpenMM 7: Rapid Development of High Performance Algorithms for Molecular Dynamics. PLOS Comput. Biol 2017, 13 (7), e1005659. 10.1371/journal.pcbi.1005659. [PubMed: 28746339]

(2). Smith JS; Isayev O; Roitberg AE ANI-1: An Extensible Neural Network Potential with DFT Accuracy at Force Field Computational Cost. Chem. Sci 2017, 8 (4), 3192–3203. 10.1039/C6SC05720A. [PubMed: 28507695]

(3). Batzner S; Musaelian A; Sun L; Geiger M; Mailoa JP; Kornbluth M; Molinari N; Smidt TE; Kozinsky B E(3)-Equivariant Graph Neural Networks for Data-Efficient and Accurate Interatomic Potentials. Nat. Commun 2022, 13 (1), 2453. 10.1038/s41467-022-29939-5. [PubMed: 35508450]

(4). Thölke P; De Fabritiis G TorchMD-NET: Equivariant Transformers for Neural Network Based Molecular Potentials. arXiv April 23, 2022. 10.48550/arXiv.2202.02541.

(5). Musaelian A; Batzner S; Johansson A; Sun L; Owen CJ; Kornbluth M; Kozinsky B Learning Local Equivariant Representations for Large-Scale Atomistic Dynamics. Nat. Commun 2023, 14 (1), 579. 10.1038/s41467-023-36329-y. [PubMed: 36737620]

(6). Batatia I; Kovács DP; Simm GNC; Ortner C; Csányi G MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields. arXiv June 15, 2022. 10.48550/arXiv.2206.07697.

(7). Simeon G; de Fabritiis G TensorNet: Cartesian Tensor Representations for Efficient Learning of Molecular Potentials. arXiv June 10, 2023. 10.48550/arXiv.2306.06482.

(8). Unke OT; Meuwly M PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges. J. Chem. Theory Comput 2019, 15 (6), 3678–3693. 10.1021/acs.jctc.9b00181. [PubMed: 31042390]

(9). Unke OT; Chmiela S; Gastegger M; Schütt KT; Sauceda HE; Müller K-R SpookyNet: Learning Force Fields with Electronic Degrees of Freedom and Nonlocal Effects. Nat. Commun 2021, 12 (1), 7273. 10.1038/s41467-021-27504-0. [PubMed: 34907176]

(10). Ko TW; Finkler JA; Goedecker S; Behler J A Fourth-Generation High-Dimensional Neural Network Potential with Accurate Electrostatics Including Non-Local Charge Transfer. Nat. Commun 2021, 12 (1), 398. 10.1038/s41467-020-20427-2. [PubMed: 33452239]

(11). Plé T; Lagardère L; Piquemal J-P Force-Field-Enhanced Neural Network Interactions: From Local Equivariant Embedding to Atom-in-Molecule Properties and Long-Range Effects. arXiv September 13, 2023. 10.48550/arXiv.2301.08734.

(12). Christensen AS; Sirumalla SK; Qiao Z; O'Connor MB; Smith DGA; Ding F; Bygrave PJ; Anandkumar A; Welborn M; Manby FR; et al. OrbNet Denali: A Machine Learning Potential for Biological and Organic Chemistry with Semi-Empirical Cost and DFT Accuracy. J. Chem. Phys 2021, 155 (20), 204103. 10.1063/5.0061990. [PubMed: 34852495]

(13). Zhou G; Lubbers N; Barros K; Tretiak S; Nebgen B Deep Learning of Dynamically Responsive Chemical Hamiltonians with Semiempirical Quantum Mechanics. Proc. Natl. Acad. Sci 2022, 119 (27), e2120333119. 10.1073/pnas.2120333119. [PubMed: 35776544]

(14). Hu F; He F; Yaron DJ Treating Semiempirical Hamiltonians as Flexible Machine Learning Models Yields Accurate and Interpretable Results. J. Chem. Theory Comput 2023, 19 (18), 6185–6196. 10.1021/acs.jctc.3c00491. [PubMed: 37705220]

(15). Rufa DA; Macdonald HEB; Fass J; Wieder M; Grinaway PB; Roitberg AE; Isayev O; Chodera JD Towards Chemical Accuracy for Alchemical Free Energy Calculations with Hybrid Physics-Based Machine Learning / Molecular Mechanics Potentials. bioRxiv July 30, 2020, p 2020.07.29.227959. 10.1101/2020.07.29.227959.

(16). Galvelis R; Varela-Rial A; Doerr S; Fino R; Eastman P; Markland TE; Chodera JD; De Fabritiis G NNP/MM: Accelerating Molecular Dynamics Simulations with Machine Learning Potentials and Molecular Mechanics. J. Chem. Inf. Model 2023. 10.1021/acs.jcim.3c00773.

(17). Talirz L; Ghiringhelli LM; Smit B Trends in Atomistic Simulation Software Usage [Article v1.0]. Living J. Comput. Mol. Sci 2021, 3 (1), 1483–1483. 10.33011/livecoms.3.1.1483.

(18). Paszke A; Gross S; Massa F; Lerer A; Bradbury J; Chanan G; Killeen T; Lin Z; Gimelshein N; Antiga L; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems; Curran Associates, Inc., 2019; Vol. 32.

(19). Devereux C; Smith JS; Huddleston KK; Barros K; Zubatyuk R; Isayev O; Roitberg AE Extending the Applicability of the ANI Deep Learning Molecular Potential to Sulfur and Halogens. J. Chem. Theory Comput 2020, 16 (7), 4192–4202. 10.1021/acs.jctc.0c00121. [PubMed: 32543858]

(20). Smith JS; Nebgen BT; Zubatyuk R; Lubbers N; Devereux C; Barros K; Tretiak S; Isayev O; Roitberg AE Approaching Coupled Cluster Accuracy with a General-Purpose Neural Network Potential through Transfer Learning. Nat. Commun 2019, 10, 2903. 10.1038/s41467-019-10827-4. [PubMed: 31263102]

(21). Eastman P; Behara PK; Dotson DL; Galvelis R; Herr JE; Horton JT; Mao Y; Chodera JD; Pritchard BP; Wang Y; et al. SPICE, A Dataset of Drug-like Molecules and Peptides for Training Machine Learning Potentials. Sci. Data 2023, 10 (1), 11. 10.1038/s41597-022-01882-6. [PubMed: 36599873]

(22). Najibi A; Goerigk L The Nonlocal Kernel in van Der Waals Density Functionals as an Additive Correction: An Extensive Analysis with Special Emphasis on the B97M-V and ωB97M-V Approaches. J. Chem. Theory Comput 2018, 14 (11), 5725–5738. 10.1021/acs.jctc.8b00842. [PubMed: 30299953]

(23). Barducci A; Bussi G; Parrinello M Well-Tempered Metadynamics: A Smoothly Converging and Tunable Free-Energy Method. Phys. Rev. Lett 2008, 100 (2), 020603. 10.1103/PhysRevLett.100.020603. [PubMed: 18232845]

(24). Tribello GA; Bonomi M; Branduardi D; Camilloni C; Bussi G PLUMED 2: New Feathers for an Old Bird. Comput. Phys. Commun 2014, 185 (2), 604–613. 10.1016/j.cpc.2013.09.018.

(25). Marinari E; Parisi G Simulated Tempering: A New Monte Carlo Scheme. Europhys. Lett 1992, 19 (6), 451. 10.1209/0295-5075/19/6/002.

(26). Hamelberg D; de Oliveira CAF; McCammon JA Sampling of Slow Diffusive Conformational Transitions with Accelerated Molecular Dynamics. J. Chem. Phys 2007, 127 (15), 155102. 10.1063/1.2789432. [PubMed: 17949218]

(27). Swendsen RH; Wang J-S Replica Monte Carlo Simulation of Spin-Glasses. Phys. Rev. Lett 1986, 57 (21), 2607–2609. 10.1103/PhysRevLett.57.2607. [PubMed: 10033814]

(28). Wang L; Friesner RA; Berne BJ Replica Exchange with Solute Scaling: A More Efficient Version of Replica Exchange with Solute Tempering (REST2). J. Phys. Chem. B 2011, 115 (30), 9431–9438. 10.1021/jp204407d. [PubMed: 21714551]

(29). Azimi S; Khuttan S; Wu JZ; Pal RK; Gallicchio E Relative Binding Free Energy Calculations for Ligands with Diverse Scaffolds with the Alchemical Transfer Method. J. Chem. Inf. Model 2022, 62 (2), 309–323. 10.1021/acs.jcim.1c01129. [PubMed: 34990555]

(30). Maier JA; Martinez C; Kasavajhala K; Wickstrom L; Hauser KE; Simmerling C ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB. J. Chem. Theory Comput 2015, 11 (8), 3696–3713. 10.1021/acs.jctc.5b00255. [PubMed: 26574453]

(31). Huang J; MacKerell AD Jr CHARMM36 All-Atom Additive Protein Force Field: Validation Based on Comparison to NMR Data. J. Comput. Chem 2013, 34 (25), 2135–2145. 10.1002/jcc.23354. [PubMed: 23832629]

(32). Shi Y; Xia Z; Zhang J; Best R; Wu C; Ponder JW; Ren P Polarizable Atomic Multipole-Based AMOEBA Force Field for Proteins. J. Chem. Theory Comput 2013, 9 (9), 4046–4063. 10.1021/ct4003702. [PubMed: 24163642]

(33). Lopes PEM; Huang J; Shim J; Luo Y; Li H; Roux B; MacKerell AD Polarizable Force Field for Peptides and Proteins Based on the Classical Drude Oscillator. J. Chem. Theory Comput 2013, 9 (12), 5430–5449. 10.1021/ct400781b. [PubMed: 24459460]

(34). Kirschner KN; Yongye AB; Tschampel SM; González-Outeiriño J; Daniels CR; Foley BL; Woods RJ GLYCAM06: A Generalizable Biomolecular Force Field. Carbohydrates. J. Comput. Chem 2008, 29 (4), 622–655. 10.1002/jcc.20820. [PubMed: 17849372]

(35). Wagner J; Thompson M; Dotson D; hyejang; SimonBoothroyd; Rodríguez-Guerra, J. Openforcefield/Openff-Forcefields: Version 2.0.0 "Sage." 2021. 10.5281/zenodo.5214478.

(36). Wang J; Wolf RM; Caldwell JW; Kollman PA; Case DA Development and Testing of a General Amber Force Field. J. Comput. Chem 2004, 25 (9), 1157–1174. 10.1002/jcc.20035. [PubMed: 15116359]

(37). Leimkuhler B; Matthews C Efficient Molecular Dynamics Using Geodesic Integration and Solvent–Solute Splitting. Proc R Soc A 2016, 472 (2189), 20160138. 10.1098/rspa.2016.0138. [PubMed: 27279779]

(38). Zhang Z; Liu X; Yan K; Tuckerman ME; Liu J Unified Efficient Thermostat Scheme for the Canonical Ensemble with Holonomic or Isokinetic Constraints via Molecular Dynamics. J. Phys. Chem. A 2019, 123 (28), 6056–6079. 10.1021/acs.jpca.9b02771. [PubMed: 31117592]

(39). Hoover WG Canonical Dynamics: Equilibrium Phase-Space Distributions. Phys. Rev. A 1985, 31 (3), 1695–1697. 10.1103/PhysRevA.31.1695.

(40). Lamoureux G; Roux B Modeling Induced Polarization with Classical Drude Oscillators: Theory and Molecular Dynamics Simulation Algorithm. J. Chem. Phys 2003, 119 (6), 3025–3039. 10.1063/1.1589749.

(41). Hahn D; Bayly C; Boby ML; Macdonald HB; Chodera J; Gapsys V; Mey A; Mobley D; Benito LP; Schindler C; et al. Best Practices for Constructing, Preparing, and Evaluating Protein-Ligand Binding Affinity Benchmarks [Article v1.0]. Living J. Comput. Mol. Sci 2022, 4 (1), 1497–1497. 10.33011/livecoms.4.1.1497. [PubMed: 36382113]

(42). Kojima S; Ohkawa H; Hirano T; Maki S; Niwa H; Ohashi M; Inouye S; Tsuji FI Fluorescent Properties of Model Chromophores of Tyrosine-66 Substituted Mutants of Aequorea Green Fluorescent Protein (GEP). Tetrahedron Lett. 1998, 39 (29), 5239–5242. 10.1016/S00404039(98)01031-4.

(43). Craggs TD Green Fluorescent Protein: Structure, Folding and Chromophore Maturation. Chem. Soc. Rev 2009, 38 (10), 2865–2875. 10.1039/b903641p. [PubMed: 19771333]

(44). Zuehlsdorff TJ; Napoli JA; Milanese JM; Markland TE; Isborn CM Unraveling Electronic Absorption Spectra Using Nuclear Quantum Effects: Photoactive Yellow Protein and Green Fluorescent Protein Chromophores in Water. J. Chem. Phys 2018, 149 (2), 024107. 10.1063/1.5025517. [PubMed: 30007372]

(45). Hutter J; Iannuzzi M; Schiffmann F; VandeVondele J Cp2k: Atomistic Simulations of Condensed Matter Systems. WIREs Comput. Mol. Sci 2014, 4 (1), 15–25. 10.1002/wcms.1159.

(46). Perdew JP; Burke K; Ernzerhof M Generalized Gradient Approximation Made Simple. Phys. Rev. Lett 1996, 77 (18), 3865–3868. 10.1103/PhysRevLett.77.3865. [PubMed: 10062328]

(47). Zhang Y; Yang W Comment on ``Generalized Gradient Approximation Made Simple''. Phys. Rev. Lett 1998, 80 (4), 890–890. 10.1103/PhysRevLett.80.890.

(48). Grimme S; Antony J; Ehrlich S; Krieg H A Consistent and Accurate Ab Initio Parametrization of Density Functional Dispersion Correction (DFT-D) for the 94 Elements H-Pu. J. Chem. Phys 2010, 132 (15), 154104. 10.1063/1.3382344. [PubMed: 20423165]

(49). Tuckerman M; Berne BJ; Martyna GJ Reversible Multiple Time Scale Molecular Dynamics. J. Chem. Phys 1992, 97 (3), 1990–2001. 10.1063/1.463137.

(50). Aradi B; Hourahine B; Frauenheim, Th. DFTB+, a Sparse Matrix-Based Implementation of the DFTB Method. J. Phys. Chem. A 2007, 111 (26), 5678–5684. 10.1021/jp070186p. [PubMed: 17567110]

(51). Jones CM; List NH; Martínez TJ Resolving the Ultrafast Dynamics of the Anionic Green Fluorescent Protein Chromophore in Water. Chem. Sci 2021, 12 (34), 11347–11363. 10.1039/D1SC02508B. [PubMed: 34667545]
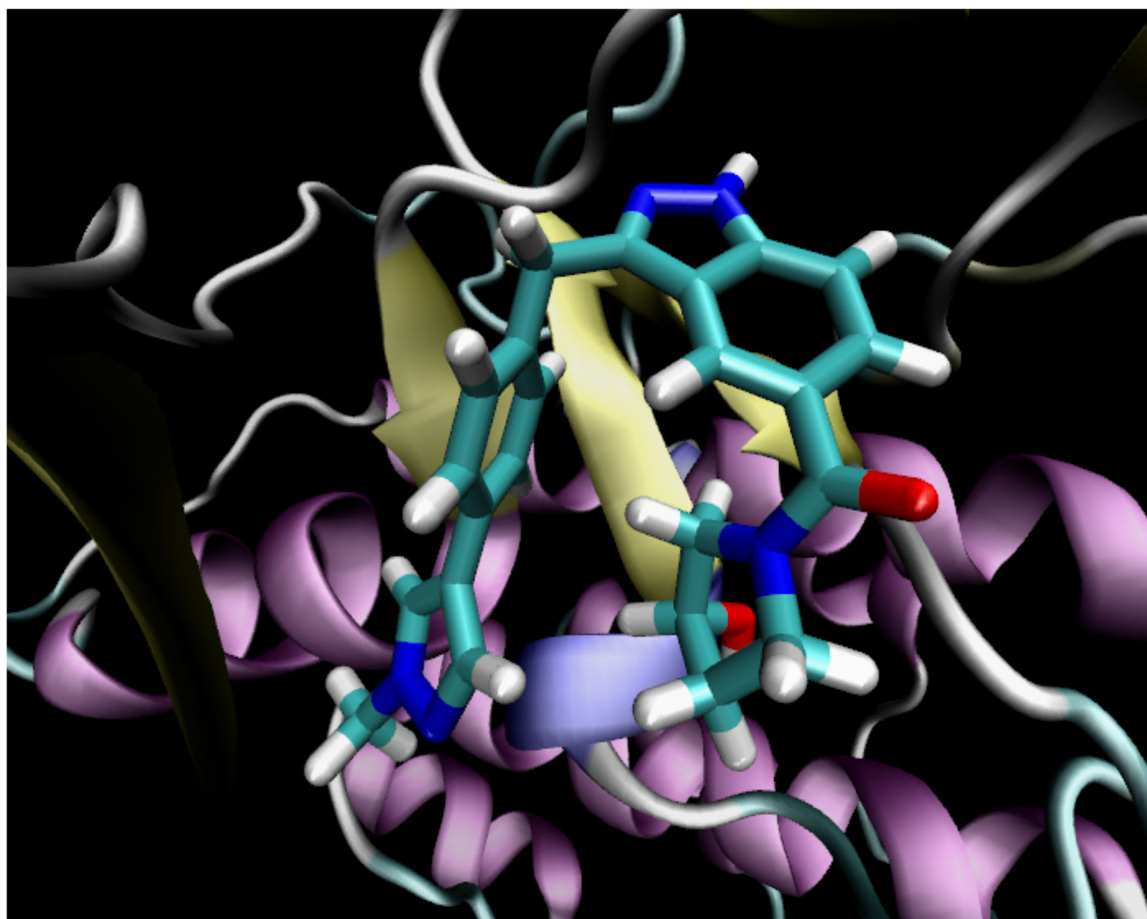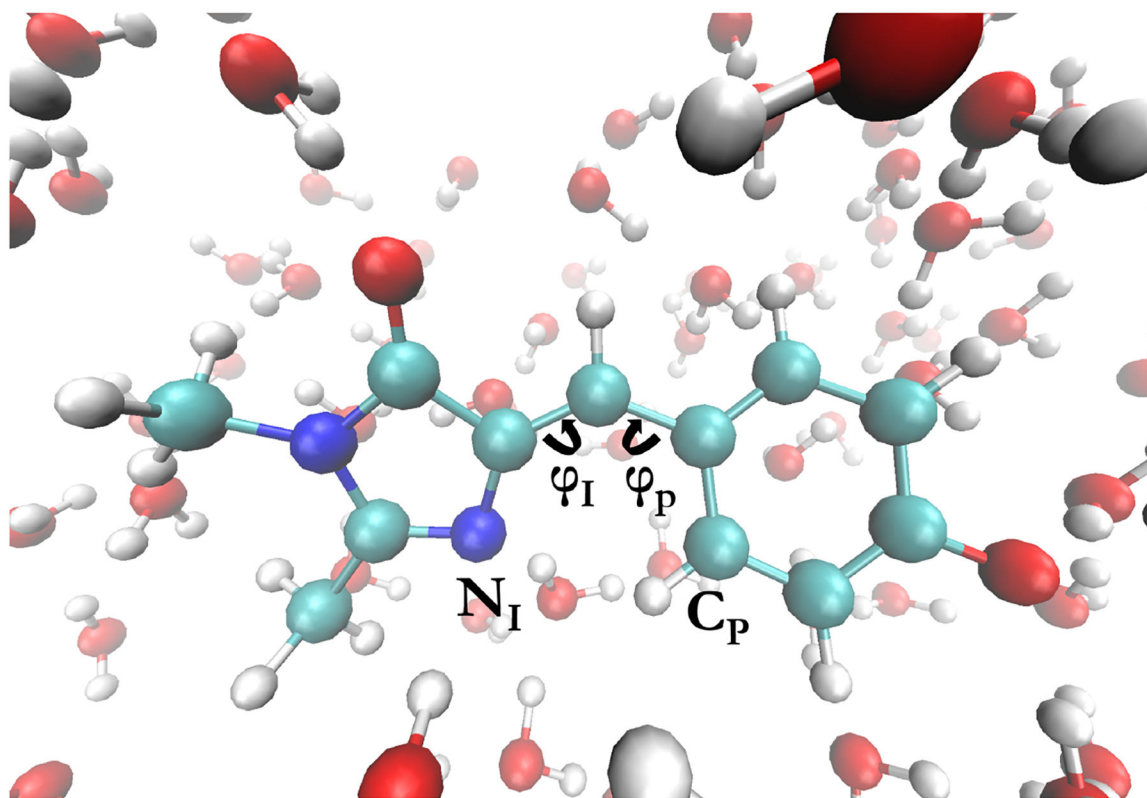
Figure 1:
The 53 atom inhibitor bound to CDK8.

Figure 2:
Anionic GFP chromophore in water. The dihedral ϕ_P is defined between the C_P atom and the carbons of the methine bridge and the dihedral ϕ_I is defined between the N_I atom and the carbons of the methine bridge.
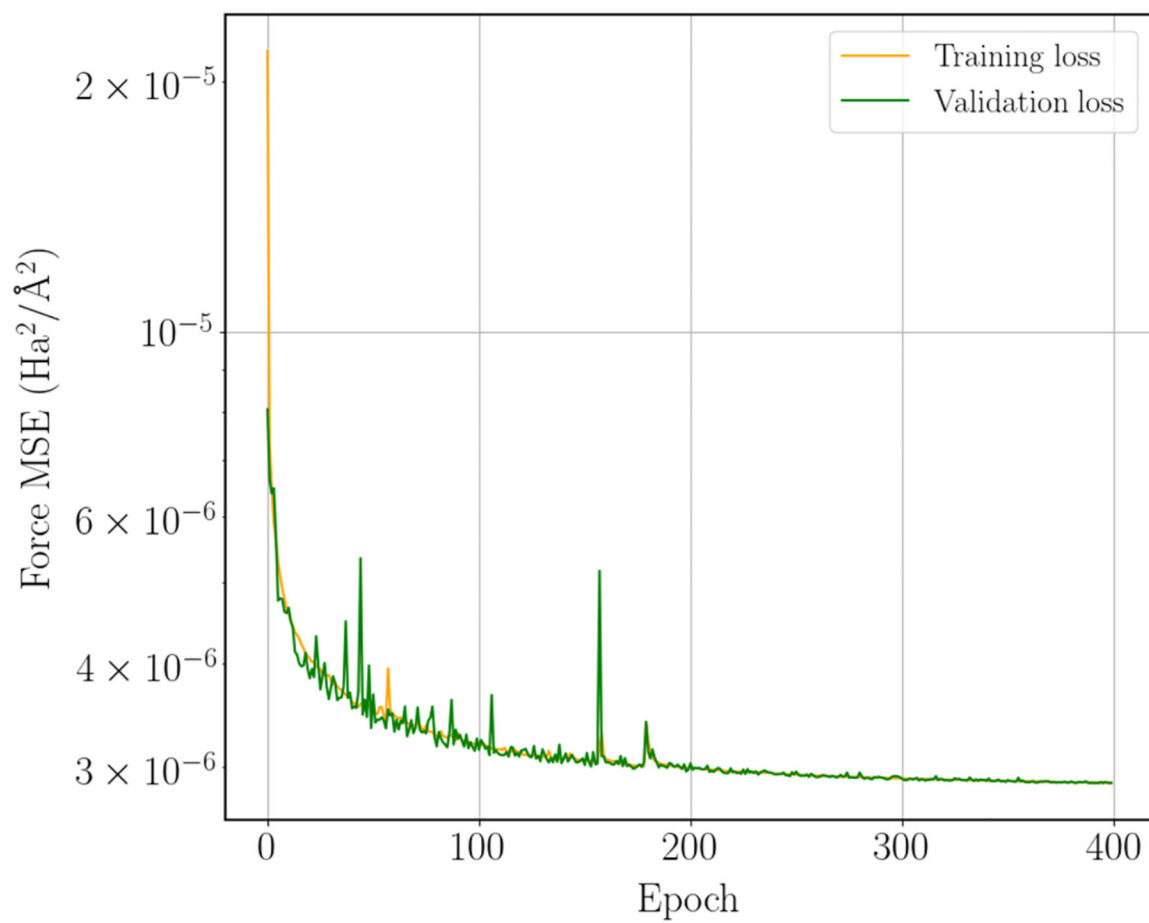
Figure 3:
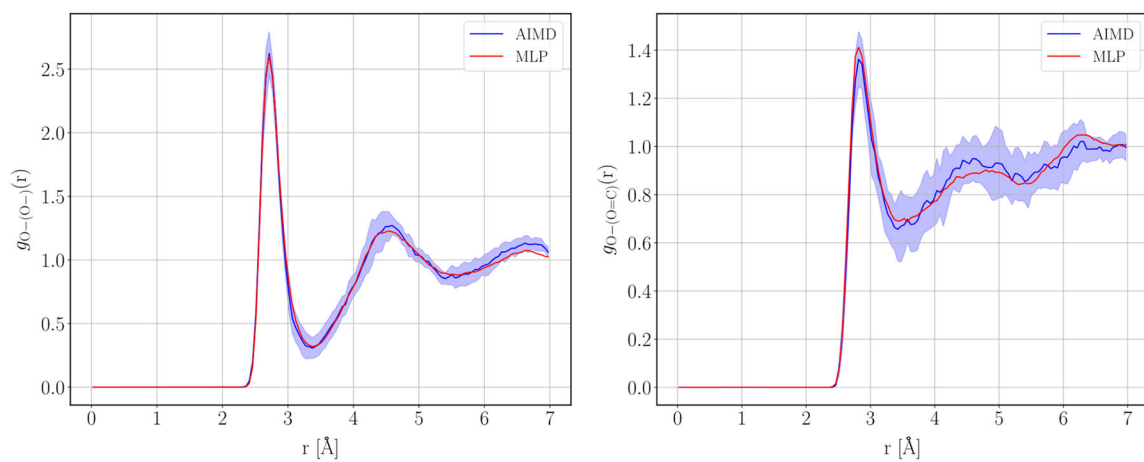Equivariant transformer training and validation loss.

Figure 4:
Anionic oxygen-oxygen RDF (left) and carbonyl oxygen-oxygen RDF (right) generated from the MLP and AIMD trajectories. The shaded region represents +/− 1 standard deviation across the 10 AIMD trajectories.
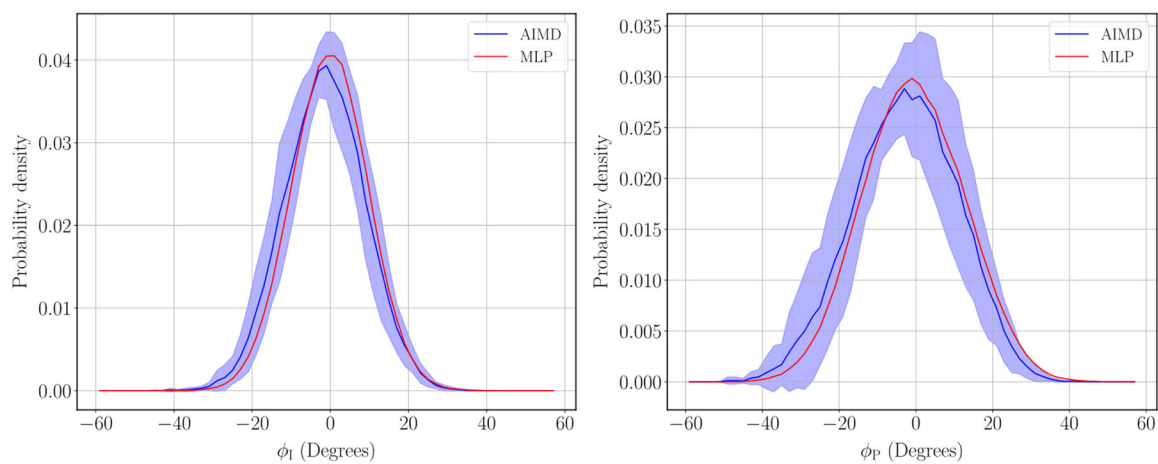
Figure 5:
Distribution of dihedral angles for ϕ_I and ϕ_P. The shaded region represents $+/-$ 1 standard deviation across the 10 AIMD trajectories.

Table 1.

The speed of the OpenCL and HIP platforms simulating three benchmark systems on an AMD V620 GPU. All benchmarks used Langevin dynamics with a 4 fs time step, constraining the lengths of bonds involving hydrogen.

Molecule	Atoms	OpenCL (ns/day)	HIP (ns/day)
DHFR	23,558	417	1031
ApoAl	92,224	192	393
Cellulose	408,609	42	91

Table 2.

The speed of simulating Satellite Tobacco Mosaic Virus (1,067,095 atoms) on one to four NVIDIA A100 GPUs connected by NVLink. All benchmarks use the CUDA platform, Langevin dynamics, a 4 fs time step, and constraints on the lengths of bonds involving hydrogen.

GPUs	Speed (ns/day)
1	32.3
2	51.5
3	61.6
4	70.9

Table 3.

The speed of simulating a 53 atom inhibitor bound to CDK8.

Potential	Implementation	Step Size (fs)	Speed (ns/day)
ANI-2x (ensemble)	TorchANI	2	9.8
ANI-2x (ensemble)	NNPOps	2	56
ANI-2x (single)	NNPOps	2	105
OpenFF 2.0.0		2	281
OpenFF 2.0.0		4	531