The Journal of Open Source Software

HOOMD-TF: GPU-Accelerated, Online Machine Learning in the HOOMD-blue Molecular Dynamics Engine

Rainier Barrett[1], **Maghesree Chakraborty**[1], **Dilnoza B Amirkulova**[1], **Heta A Gandhi**[1], **Geemi P Wellawatte**[2], **and Andrew D White**[1]

1 University of Rochester Chemical Engineering Department, Rochester, New York, United States of America **2** University of Rochester Chemistry Department, Rochester, New York, United States of America

DOI: 10.21105/joss.02367

Software

- Review
- Repository
- Archive

Editor: Richard Gowers

Reviewers:

- @malramsay64
- @rmeli

Submitted: 27 May 2020
Published: 28 July 2020

Statement of Need

Machine learning (ML) is emerging as an essential tool in the molecular modeling community. The most mature applications are in batch learning with data generated from molecular modeling calculations. For example, one can run a molecular dynamics (MD) or density functional theory (DFT) simulation to completion, from which the trajectory is then used as input data to train a deep neural network that reproduces the energetics at a fraction of the computational cost. Some recent examples include an energy-conserving force-field learned with a custom gradient-domain model (Chmiela et al., 2017), DFT-based neural network force-fields (J. S. Smith et al., 2017a), and a neural network coarse-grained potential (Wang et al., 2019). Other applications include the use of ML methods for collective variable (CV) calculation (Trapl et al., 2019) and enhanced sampling (Wang, Lamim Ribeiro, & Tiwary, 2020) for MD simulations. A limitation of these methods is that they treat the molecular modeling calculations as a static dataset, whereas molecular modeling calculations can be treated as interrogative functions, opening the door to methods like reinforcement learning or active learning. Thus there is a clear limitation of existing implementations due to their sequential nature of going from calculation to ML. Another practical issue is that neural network force-field implementations often duplicate standard ML frameworks, preventing them from keeping progress with state-of-the art methods. For example Rupp, Tkatchenko, Müller, & Anatole Von Lilienfeld (2012) and Botu & Ramprasad (2015) are benchmark works in this field and custom implementations. This limits the scope and speed of translating ML advances to the molecular modeling community.

This need has led us to develop HOOMD-TF, a flexible direct integration of a standard ML library and standard molecular simulation framework that maintains GPU acceleration. Our goals are to improve the reproducibility of ML methods in molecular simulation, ease translation of ML advances, and remove the need for sequential simulation and ML. This should enable active learning, reinforcement learning, and online learning of molecular simulations.

There are other applications focusing on ML in molecular modeling, such as DeepChem and Pyzer-Knapp, Li, & Aspuru-Guzik (2015), which are largely concerned with property prediction and representation. There are also similar works to HOOMD-TF called OpenMM-NN (Eastman, 2018), which allows the use of pre-trained TensorFlow models in OpenMM (Eastman et al., 2013), and TorchANI (Gao, Ramezanghorbani, Isayev, Smith, & Roitberg, 2020), which uses PyTorch (Paszke, Gross, Chintala, & Chanan, 2016) for similar purposes. In contrast, HOOMD-TF fills the niche of online model training, while also allowing pre-trained model imports in MD simulation, coarse-grained force-field learning, collective variable calculation and manipulation, and force-field biasing.

Summary

The HOOMD-TF package pairs the TensorFlow ML library (Abadi et al., 2015) with the HOOMD-blue simulation engine (Anderson, Glaser, & Glotzer, 2020) to allow for flexible online ML and tensor calculations during HOOMD-blue simulations. Since both TensorFlow and HOOMD-blue are GPU-accelerated, HOOMD-TF was designed with a GPU-GPU communication scheme that minimizes latency between GPU memory to preserve execution speed.

HOOMD-TF enables online ML in MD simulations with the support of the suite of tools available through TensorFlow. It can be used for force matching, calculation of arbitrary collective variables, force-field biasing or learning using said CVs, and analysis using tensor calculations. These tasks can be performed either online during a simulation or offline using a saved trajectory. This is accomplished by using TensorFlow tensors filled with particle positions and neighbor lists from HOOMD-blue. This also allows the use of TensorFlow's derivative propagation to perform biasing with arbitrary CVs, provided that they can be expressed as a tensor operation of either the neighbor list or particle positions. Another application of this software is learning coarse-grained force-fields with either neural networks or other ML models. The ability to run force matching calculations online makes the coarse-graining workflow straightforward in HOOMD-TF. Since HOOMD-blue can use external force-fields and TensorFlow can learn as the simulation is running, learning can be terminated as soon as the force-matching algorithm converges, requiring only one simulation iteration. Contrast this with a popular force-matching package, VOTCA (Rühle, Junghans, Lukyanov, Kremer, & Andrienko, 2009), which uses an iterative approach. See, for example, Jadrich, Lindquist, & Truskett (2017).

HOOMD-TF uses TensorFlow to save and load models, and is therefore compatible with pre-trained TensorFlow models. TensorFlow's TensorBoard utility can also be used to track and examine model training and performance. HOOMD-TF can be used independent of HOOMD-blue by using trajectories via the MDAnalysis framework (Gowers et al., 2016; Michaud-Agrawal & Beckstein, 2011). This allows for previously-trained TensorFlow models to be used on trajectories that were produced by other MD engines, analysis of new CVs from a previously-run simulation, and training of models from trajectories. This offline execution scheme is functionally similar to TorchANI (Gao et al., 2020). TorchANI uses PyTorch (Paszke et al., 2016) rather than TensorFlow to implement the ANI deep learning models (J. S. Smith et al., 2017b), with many of the same advantages provided by HOOMD-TF. TorchANI is not an MD engine, so it has less support for specific features like neighbor lists or particle mesh Ewald summation.

Overall, HOOMD-TF makes online ML in MD simulations possible with little additional effort, and eases the use of TensorFlow models on MD trajectories for both machine learning and analysis. The ability to tightly integrate trained ML models in HOOMD-TF can enable their use in simulations by removing the need for custom implementations and improve reproducibility in the field. The online functionality of HOOMD-TF enables the use of simulations as interrogable models rather than static data generators, allowing direct use in an active and/or reinforcement learning framework. TensorFlow computation graphs allow for transparent and simple model designation with a high degree of customizability, replicability, and efficiency.

Accessing the Software

HOOMD-TF is freely available under the MIT license on github. The documentation is hosted on readthedocs.io.

Acknowledgements

We would like to thank Joshua Anderson and Jens Glaser for instructive conversations about the HOOMD-blue architecture. We thank the Center for Integrated Research Computing (CIRC) at the University of Rochester for providing computational resources and technical support. This work was supported by the National Science Foundation (CBET-1751471 and CHE-1764415).

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., et al. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Retrieved from https://www.tensorflow.org/

Anderson, J. A., Glaser, J., & Glotzer, S. C. (2020). HOOMD-blue: A Python package for high-performance molecular dynamics and hard particle Monte Carlo simulations. *Computational Materials Science*, *173*, 109363. doi:10.1016/j.commatsci.2019.109363

Botu, V., & Ramprasad, R. (2015). Adaptive Machine Learning Framework to Accelerate Ab Initio Molecular Dynamics. *Int. J. Quantum Chem.*, (115), 1074–1083. doi:10.1002/qua.24836

Chmiela, S., Tkatchenko, A., Sauceda, H. E., Poltavsky, I., Schütt, K. T., & Müller, K.-R. (2017). Machine learning of accurate energy-conserving molecular force fields. *Sci. Adv.*, *3*(5), e1603015. doi:10.1126/sciadv.1603015

Eastman, P. (2018). OpenMM neural network plugin. Retrieved from https://github.com/pandegroup/openmm-nn

Eastman, Peter, Friedrichs, Mark S., Chodera, John D., Radmer, Randall J., Bruns, Christopher M., Ku, Joy P., Beauchamp, Kyle A., et al. (2013). OpenMM 4: A Reusable, Extensible, Hardware Independent Library for High Performance Molecular Simulation. *Journal of Chemical Theory and Computation*, *9*(1), 461–469. doi:10.1021/ct300857j

Gao, X., Ramezanghorbani, F., Isayev, O., Smith, J. S., & Roitberg, A. E. (2020). TorchANI: A Free and Open Source PyTorch Based Deep Learning Implementation of the ANI Neural Network Potentials. *Journal of Chemical Information and Modeling*, acs.jcim.0c00451. doi:10.1021/acs.jcim.0c00451

Gowers, Linke, Barnoud, Reddy, Melo, Seyler, Domański, et al. (2016). MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations. In Sebastian Benthall & Scott Rostrup (Eds.), *Proceedings of the 15th Python in Science Conference* (pp. 98–105). doi:10.25080/Majora-629e541a-00e

Jadrich, R. B., Lindquist, B. A., & Truskett, T. M. (2017). Probabilistic inverse design for self-assembling materials. *Journal of Chemical Physics*, *146*(18), 184103. doi:10.1063/1.4981796

Michaud-Agrawal, D., N., & Beckstein, O. (2011). MDAnalysis: A toolkit for the analysis of molecular dynamics simulations. *Journal of Computational Chemistry*, *32*(10), 2319–2327. doi:10.1002/jcc.21787

Paszke, A., Gross, S., Chintala, S., & Chanan, G. (2016). PyTorch. Retrieved from https://github.com/pytorch/pytorch

Pyzer-Knapp, E. O., Li, K., & Aspuru-Guzik, A. (2015). Learning from the harvard clean energy project: The use of neural networks to accelerate materials discovery. *Advanced Functional Materials*, *25*(41), 6495–6502. doi:10.1002/adfm.201501919

Rupp, M., Tkatchenko, A., Müller, K.-R., & Anatole Von Lilienfeld, O. (2012). Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Phys. Rev. Lett.*, *108*, 058301. doi:10.1103/PhysRevLett.108.058301

Rühle, V., Junghans, C., Lukyanov, A., Kremer, K., & Andrienko, D. (2009). Versatile object-oriented toolkit for coarse-graining applications. *Journal of Chemical Theory and Computation*, *5*(12), 3211–3223. doi:10.1021/ct900369w

Smith, J. S., Isayev, O., & Roitberg, A. E. (2017a). ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chem. Sci.*, *8*(4), 3192–3203. doi:10.1039/C6SC05720A

Smith, J. S., Isayev, O., & Roitberg, A. E. (2017b). ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chemical Science*, *8*(4), 3192–3203. doi:10.1039/C6SC05720A

Trapl, D., Horvacanin, I., Mareska, V., Ozcelik, F., Unal, G., & Spiwok, V. (2019). Anncolvar: Approximation of Complex Collective Variables by Artificial Neural Networks for Analysis and Biasing of Molecular Simulations. *Frontiers in Molecular Biosciences*, *6*(APR), 25. doi:10.3389/fmolb.2019.00025

Wang, J., Olsson, S., Wehmeyer, C., Pérez, A., Charron, N. E., De Fabritiis, G., Noé, F., et al. (2019). Machine Learning of Coarse-Grained Molecular Dynamics Force Fields. *ACS Central Science*, *5*(5), 755–767. doi:10.1021/acscentsci.8b00913

Wang, Y., Lamim Ribeiro, J. M., & Tiwary, P. (2020, April). Machine learning approaches for analyzing and enhancing molecular dynamics simulations. Elsevier Ltd. doi:10.1016/j.sbi.2019.12.016