The Journal of Chemical Physics EDITORIAL pubs.aip.org/aip/jcp

Guest editorial: Special Topic on software for atomistic machine learning

Cite as: J. Chem. Phys. 161, 060401 (2024); doi: 10.1063/5.0228461
Submitted: 12 July 2024 • Accepted: 18 July 2024 •
Published Online: 9 August 2024

Matthias Rupp,[1,a)] Emine Küçükbenli,[2,b)] and Gábor Csányi[3,c)]

AFFILIATIONS

[1] Luxembourg Institute of Science and Technology, L-4362 Esch-sur-Alzette, Luxembourg
[2] John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts 02138, USA and Nvidia Corporation, Santa Clara, California 95051, USA
[3] Engineering Laboratory, University of Cambridge, Cambridge CB2 1PZ, United Kingdom

Note: This paper is part of the JCP Special Topic on Software for Atomistic Machine Learning.
[a)] **Author to whom correspondence should be addressed:** mrupp@mrupp.info
[b)] ekucukbenli@nvidia.com
[c)] gc121@cam.ac.uk

https://doi.org/10.1063/5.0228461

I. INTRODUCTION

Welcome to the Journal of Chemical Physics' Special Topic on Software for Atomistic Machine Learning. For some years now, search engines have been dominating our online experience and have essentially overtaken libraries, whether physical or digital, as the means to find information we are looking for. Most readers of an original research article find it by citation or direct search, and not by browsing journal volumes. Given this, one might wonder what the utility of a Special Topic issue of a scientific journal might be.

However, publishing papers on scientific software has traditionally been somewhat neglected, with few go-to journals for publishing, such as the Journal of Open Source Software or Computer Physics Communications. Typical published software papers tend to discuss relatively mature software packages. In this context, the Journal of Chemical Physics' initiative[1] to support software publications is especially welcome. Given the huge activity currently taking place across many sub-fields and communities in new software development for atomistic machine learning (ML), this landscape is changing fast. Arguably, many papers in this Special Topic issue might not have been written if it were not for the impetus provided by this Special Topic issue.

Beyond their individual value regarding specific software packages, these papers as a collection provide a snapshot at this moment in time of the kinds of tools that people use and the goals they set themselves and achieve for the software implementations of their methods. Table I presents an overview of the 28 invited and contributed articles.[2–29] Of these, 18 (64%) deal directly with machine-learning interatomic potentials (MLIPs). The other ten articles cover a broad range of subjects, ranging from sampling to dataset repositories and workflows.

In the following, we give an overview of these contributions.

II. CONTRIBUTIONS

Since their beginnings in the 1980s and 1990s, MLIPs have undergone tremendous development and now constitute a highly active field of research. Some modern MLIPs can predict forces with an accuracy close to the underlying *ab initio* reference method for atomistic systems with many chemical elements and millions of atoms while still providing orders of magnitude of acceleration. These capabilities have increasingly enabled scientific applications using MLIPs that would not otherwise have been possible.

Consequently, there is a trend to **directly integrate MLIPs into molecular dynamics codes**. In this Special Topic, four contributions describe the integration of (a) neuro-evolution potentials into GPUMD (Graphics Processing Units Molecular Dynamics), including improved featurization, GPU code, active learning, and supporting Python packages gpyumd, calorine, and pynep;[3] (b) PhysNet into CHARMM (Chemistry at HARvard Macromolecular Mechanics) via a new MLpot extension of the pyCHARMM interface, with *para*-chlorophenol as an example;[12] (c) general MLIPs into CASTEP (CAmbridge Serial Total Energy Package), including active learning, using the example of a Gaussian Approximation Potential (GAP)/smooth overlap of atomic positions (SOAP) model;[16] and (d)

TABLE I. Overview of contributions to the Special Topic. See the nomenclature for acronyms.

References	Package	ML	Language	License	Data	Comments
2	AGOX	⋯	Python	GPL3	Pt_{14}/Au(100)	ASE; structure search
3	GPUMD	ANN	C++	GPL3	Si, C, MD17	CUDA; neuro-evolution potentials
4	QML-lightning	GPR	Python	MIT	QM9, MD17, 3PBA	PyTorch; FCHL19, GPR, RFF, SORF
5	PESPIP	LR	Mathematica, C++, Fortran, Perl, Python	⋯	H_2O, hydrocarbons	Permutationally invariant polynomials
6	⋯	GPR	⋯	⋯	Pt nanoparticles	GAP/SOAP application
7	SchNetPack2	ANN	Python	MIT	QM9, MD17, C_2H_6O	PyTorch; MD, FieldSchnet, PaiNN
8	ænet-PyTorch	ANN	Python	MIT	TiO_2, LiMoNiTiO, amorphous Li_xSi	PyTorch; atomic energy network
9	DScribe	⋯	Python, C++	Apache2	$CsPb(Cl/Br)_3$, Cu clusters	Atomistic featurization
10	mlcolvar	DR	Python	MIT	Alanine dipeptide, aldol reaction, chignolin	Dimensionality reduction, collective variables, enhanced sampling
11	AGOX	⋯	Python	GPL3	Rutile $SnO_2(110)$–(4×1), olivine $(Mg_2SiO_4)_4$	ASE; structure generation
12	CHARMM	ANN	Python	⋯	*para*-chlorophenol	PhysNet integration
13	AL4GAP	GPR	Python	MIT	Molten salts	Ensemble active learning, GAP
14	XPOT	BO	Python	GPL2	Si, Cu, C, Ni, Li, Mo, Ge	Hyperparameter optimization
15	PES-learn	ANN	Python	BSD3	Methanol, HCOOH	Benchmark multi-fidelity approaches
16	CASTEP	GPR	Python	CAL	Al_2O_3, Si, a-C	Active-learning MLIP for CASTEP
17	q-pac	GPR	Python	MIT	QM9, ZnO, ZnO_2	Kernel charge equilibration
18	DeePMD-kit	ANN	Python, C/C++	LGPL3	H_2O, Cu, HEA, OC2M, SPICE	TensorFlow; deep potentials
19	sphericart	⋯	C++, Python	Apache2	⋯	PyTorch; fast spherical harmonics
20	MLIP-3	NLR	C++	BSD	Cu(111)	Moment tensor potentials
21	PANNA2	ANN	Python	MIT	rMD17, C, NaCl clusters	TensorFlow; ANN MLIP training
22	DeepQMC	ANN	Python	MIT	NH_3, CO, N_2, cyclobutadiene, reactions; ScO, TiO, VO, CrO	JAX; variational quantum Monte Carlo
23	SISSO++	SR	C++, Python	Apache2	⋯	Symbolic regression
24	wfl, ExPyRe	⋯	Python	GPL2	⋯	ASE-based workflows
25	EDDP	ANN	Fortran, Julia	GPL2, MIT	C, Pb, ScH_{12}, $Zn(CN)_2$	Ephemeral data-derived potentials
26	ColabFit Exchange	⋯	⋯	⋯	Many datasets	Dataset repository
27	ACEpotentials.jl	LR	Julia	MIT	Six elements, H_2O, $AlSi_{10}$, polyethylene glycol, $CsPbBr_3$	Linear GPR/ACE MLIPs
28	glp	AD	Python	MIT	SnSe	JAX; auto-differentiation, heat flux
29	QUIP	GPR	Fortran, Python, C	GPL2, ASL	Si, core e binding energies, MoNbTaVW	GAP MLIPs, MPI parallelization

ephemeral data-derived potentials with AIRSS (*Ab Initio* Random Structure Search).[25]

An important milestone in MLIP development was the introduction of **artificial neural network MLIPs** that were able to efficiently handle large ("high-dimensional") atomistic systems by Behler and Parrinello.[30] Six contributions in this Special Topic present MLIPs related to Behler–Parrinello networks: (a) neuroevolution potentials;[3] (b) the atomic energy network (ænet) in a PyTorch implementation for GPU support;[8] (c) deep potentials via DeePMD-kit, recent improvements including attention-based features, learning dipoles and polarizabilities, long-range interactions, model compression, and GPU acceleration;[18] (d) multi-layer-perceptron-based MLIPs via PANNA (Properties from Artificial Neural Network Architectures), with improved GPU support based on TensorFlow and long-range electrostatic interactions through a variational charge equilibration scheme;[21] and (e) ephemeral data-derived potentials (EDDPs) for atomistic structure prediction, including ensemble-based uncertainties.[25]

Message-passing neural networks allow the exchange of information between atoms beyond their local environments by repeatedly passing messages between them. Two contributions provide such MLIPs: (a) the SchNetPack2 library provides improved support functionality, including data sparsity, equivariance, and PyTorch-based MD, and provides four MLIPs: SchNet and FieldSchnet (external fields) and PaiNN and SO3net (equivariance);[7] (b) the existing PhysNet MLIP is integrated into CHARMM.[12]

Kernel-based learning is another ML approach that many MLIPs employ. Gaussian process regression (GPR), in particular, has been frequently used since the introduction of Gaussian Approximation Potentials (GAPs).[31] Five contributions in this Special Topic deal with kernel-based MLIPs: (a) the quantum machine learning (QML)-lightning package provides GPU-accelerated sparse approximate GPR and representations (random features, FCHL19);[4] (b) AL4GAP provides ensemble-based active learning for GAP MLIPs to study charge-neutral molten-salt mixtures,[13] (c) the q-pac package implements kernel charge equilibration based on sparse GPR for long-range electrostatic interactions, non-local charge transfer, and energetic response to external fields;[17] (d) the quantum mechanics and interatomic potentials (QUIP) package allows training and deployment of GAP models, recent additions including distributed training via the Message Passing Interface (MPI) and compressed features;[29] and (e) an application study develops a GAP/SOAP MLIP to obtain the pressure–temperature phase diagram of Pt and to simulate the spontaneous crystallization of a large Pt nanoparticle.[6]

Other ML approaches can be used to develop MLIPs, notably linear regression. Three contributions describe such MLIPs: (a) the ACEpotentials.jl Julia package provides MLIPs based on linear GPR and the atomic cluster expansion (ACE) representation, including uncertainties and active learning;[27] (b) the PESPIP (Potential Energy Surface Permutationally Invariant Polynomials) package provides MLIPs based on permutationally invariant polynomials in Morse-transformed interatomic distances, including their optimization;[5] and (c) the MLIP-3 package provides moment tensor potentials, including fragment-based active learning for large simulation cells.[20]

Besides the MLIPs themselves, seven contributions provide **auxiliary tooling and analysis** that focus on specific aspects of MLIPs but are not specific to one MLIP: (a) the DScribe library provides many atomistic representations and has been extended to include Valle–Oganov materials fingerprints and derivatives for all representations;[9] (b) the sphericart package implements efficient real-valued spherical harmonics, a key ingredient of many representations for MLIPs, including stable Cartesian derivatives;[19] (c) the XPOT (Cross-Platform Optimizer for Potentials) package provides hyperparameter optimization for MLIPs;[14] (d) PES-Learn benchmarks four approaches to train neural-network MLIPs on data with different levels of fidelity (e.g., low and high accuracy);[15] (e) the wfl (Workflow) and ExPyRe (Execute Python Remotely) packages provide workflow management routines tailored for atomistic simulations and MLIP development;[24] (f) the ColabFit Exchange repository hosts hundreds of diverse datasets of atomistic systems in extended XYZ format for MLIP benchmarking and development;[26] and (g) the glp package demonstrates how to use automatic differentiation to efficiently obtain forces, stress, and heat flux for message-passing MLIPs.[28]

The remaining five contributions span a wide range of **other areas**: (a) the mlcolvar library implements multiple dimensionality reduction methods to identify collective variables for analysis and enhanced sampling in MD simulations, including an interface to the PLUMED (PLUgin for MolEcular Dynamics) software;[10] (b) the AGOX (Atomistic Global Optimization X) package enables developing global optimization algorithms for atomistic structure search, including random search, basin hopping, evolutionary algorithms, and global optimization with first-principles energy expressions (GOFEE);[2] (c) the same AGOX package also includes structure generation based on local optimization in "complementary energy" landscapes (oversmoothed MLIPs), favoring structures with fewer distinct local motifs;[11] (d) the DeepQMC package provides a framework for neural network-based variational quantum Monte Carlo methods, including PauliNet, FermiNet, and DeepErwin;[22] and (e) the SISSO++ (Sure Independence Screening and Sparsifying Operator) software offers symbolic regression, including recent improvements in expression representation, support for units, nonlinear parametrization, and the solver algorithm.[23]

III. SUMMARY

This Special Topic on Software for Atomistic Machine Learning contains 28 invited and contributed articles. They range from MLIPs based on neural networks, kernel models, and linear regression, as well as their integration into MD codes, to auxiliary tooling, structure search, dimensionality reduction, quantum Monte Carlo methods, and symbolic regression. We hope you enjoy reading this community effort at capturing the state of the field at this moment.

The Journal of Chemical Physics encourages and welcomes submissions of original articles describing software implementations relevant to the broad remit of the journal.

NOMENCLATURE

Glossary

a-C	amorphous carbon
AD	automatic differentiation

ACE	atomic cluster expansion
AGOX	Atomistic Global Optimization X
AIRSS	*Ab Initio* Random Structure Search
AL	active learning
ANN	artificial neural network
ASE	Atomic Simulation Environment
ASL	Academic Software License
BO	Bayesian optimization
BSD	Berkeley Software Distribution
CAL	CASTEP Academic License
CASTEP	CAmbridge Serial Total Energy Package
CHARMM	Chemistry at HARvard Macromolecular Mechanics
CUDA	Compute Unified Device Architecture
DR	dimensionality reduction
EDDP	ephemeral data-derived potential
FCHL19	Faber, Christensen, Huang, Lilienfeld 2019
GAP	Gaussian approximation potential
GOFEE	global optimization with first-principles energy expressions
GPL	General Public License
GPR	Gaussian process regression
GPU	graphics processing unit
GPUMD	Graphics Processing Units Molecular Dynamics
JAX	Just After eXecution
LGPL	Lesser General Public License
LR	linear regression
MIT	Massachusetts Institute of Technology
MD	molecular dynamics
ML	machine learning
MLIP	machine-learning interatomic potential
MPI	message-passing interface
NLR	non-linear regression
PaiNN	Polarizable Atom Interaction Neural Network
PANNA	Properties from Artificial Neural Network Architectures
PES	potential energy surface
PIP	permutationally invariant polynomial
PLUMED	PLUgin for Molecular Dynamics
QML	quantum machine learning
QUIP	QUantum mechanics and Interatomic Potentials
RFF	random Fourier features
SISSO	Sure Independence Screening and Sparsifying Operator
SOAP	smooth overlap of atomic positions
SORF	structured orthogonal randomized features
SR	symbolic regression
XPOT	Cross-Platform Optimizer for Potentials

REFERENCES

[1]C. D. Sherrill, D. E. Manolopoulos, T. J. Martínez, M. Ceriotti, and A. Michaelides, "Chemical physics software," J. Chem. Phys. **155**(1), 010401 (2021).

[2]M.-P. V. Christiansen, N. Rønne, and B. Hammer, "Atomistic global optimization X: A Python package for optimization of atomistic structures," J. Chem. Phys. **157**(5), 054701 (2022).

[3]Z. Fan, Y. Wang, P. Ying, K. Song, J. Wang, Y. Wang, Z. Zeng, K. Xu, E. Lindgren, J. M. Rahm, A. J. Gabourie, J. Liu, H. Dong, J. Wu, Y. Chen, Z. Zhong, J. Sun, P. Erhart, Y. Su, and T. Ala-Nissila, "GPUMD: A package for constructing accurate machine-learned potentials and performing highly efficient atomistic simulations," J. Chem. Phys. **157**(11), 114801 (2022).

[4]N. J. Browning, F. A. Faber, and O. Anatole von Lilienfeld, "GPU-accelerated approximate kernel method for quantum machine learning," J. Chem. Phys. **157**(21), 214801 (2022).

[5]P. L. Houston, C. Qu, Q. Yu, R. Conte, A. Nandi, J. K. Li, and J. M. Bowman, "PESPIP: Software to fit complex molecular and many-body potential energy surfaces with permutationally invariant polynomials," J. Chem. Phys. **158**(4), 044109 (2023).

[6]J. Kloppenburg, L. B. Pártay, H. Jónsson, and M. A. Caro, "A general-purpose machine learning Pt interatomic potential for an accurate description of bulk, surfaces, and nanoparticles," J. Chem. Phys. **158**(13), 134704 (2023).

[7]K. T. Schütt, S. S. P. Hessmann, N. W. A. Gebauer, J. Lederer, and M. Gastegger, "SchNetPack 2.0: A neural network toolbox for atomistic machine learning," J. Chem. Phys. **158**(14), 144801 (2023).

[8]J. López-Zorrilla, X. M. Aretxabaleta, I. W. Yeu, I. Etxebarria, H. Manzano, and N. Artrith, "ænet-PyTorch: A GPU-supported implementation for machine learning atomic potentials training," J. Chem. Phys. **158**(16), 164105 (2023).

[9]J. Laakso, L. Himanen, H. Homm, E. V. Morooka, M. O. J. Jäger, M. Todorović, and P. Rinke, "Updates to the DScribe library: New descriptors and derivatives," J. Chem. Phys. **158**(23), 234802 (2023).

[10]L. Bonati, E. Trizio, A. Rizzi, and M. Parrinello, "A unified framework for machine learning collective variables for enhanced sampling simulations: mlcolvar," J. Chem. Phys. **159**(1), 014801 (2023).

[11]A. M. Slavensky, M.-P. V. Christiansen, and B. Hammer, "Generating candidates in global optimization algorithms using complementary energy landscapes," J. Chem. Phys. **159**(2), 024123 (2023).

[12]K. Song, S. Käser, K. Töpfer, L. I. Vazquez-Salazar, and M. Meuwly, "PhysNet meets CHARMM: A framework for routine machine learning/molecular mechanics simulations," J. Chem. Phys. **159**(2), 024125 (2023).

[13]J. Guo, V. Woo, D. A. Andersson, N. Hoyt, M. Williamson, I. Foster, C. Benmore, N. E. Jackson, and G. Sivaraman, "AL4GAP: Active learning workflow for generating DFT-SCAN accurate machine-learning potentials for combinatorial molten salt mixtures," J. Chem. Phys. **159**(2), 024802 (2023).

[14]D. F. Thomas du Toit and V. L. Deringer, "Cross-platform hyperparameter optimization for machine learning interatomic potentials," J. Chem. Phys. **159**(2), 024803 (2023).

[15]S. M. Goodlett, J. M. Turney, and H. F. Schaefer III, "Comparison of multifidelity machine learning models for potential energy surfaces," J. Chem. Phys. **159**(4), 044111 (2023).

[16]T. K. Stenczel, Z. El-Machachi, G. Liepuoniute, J. D. Morrow, A. P. Bartók, M. I. J. Probert, G. Csányi, and V. L. Deringer, "Machine-learned acceleration for molecular dynamics in CASTEP," J. Chem. Phys. **159**(4), 044803 (2023).

[17]M. Vondrák, K. Reuter, and J. T. Margraf, "q-pac: A Python package for machine learned charge equilibration models," J. Chem. Phys. **159**(5), 054109 (2023).

[18]J. Zeng, D. Zhang, D. Lu, P. Mo, Z. Li, Y. Chen, M. Rynik, L. Huang, Z. Li, S. Shi, Y. Wang, H. Ye, P. Tuo, J. Yang, Y. Ding, Y. Li, D. Tisi, Q. Zeng, H. Bao, Y. Xia, J. Huang, K. Muraoka, Y. Wang, J. Chang, F. Yuan, S. L. Bore, C. Cai, Y. Lin, B. Wang, J. Xu, J.-X. Zhu, C. Luo, Y. Zhang, R. E. A. Goodall, W. Liang, A. K. Singh, S. Yao, J. Zhang, R. Wentzcovitch, J. Han, J. Liu, W. Jia, D. M. York, W. E, R. Car, L. Zhang, and H. Wang, "DeePMD-kit v2: A software package for deep potential models," J. Chem. Phys. **159**(5), 054801 (2023).

[19]F. Bigi, G. Fraux, N. J. Browning, and M. Ceriotti, "Fast evaluation of spherical harmonics with sphericart," J. Chem. Phys. **159**(6), 064802 (2023).

[20]E. Podryabinkin, K. Garifullin, A. Shapeev, and I. Novikov, "MLIP-3: Active learning on atomic environments with moment tensor potentials," J. Chem. Phys. **159**(8), 084112 (2023).

[21]F. Pellegrini, R. Lot, Y. Shaidu, and E. Küçükbenli, "PANNA 2.0: Efficient neural network interatomic potentials and new architectures," J. Chem. Phys. **159**(8), 084117 (2023).

[22]Z. Schätzle, P. B. Szabó, M. Mezera, J. Hermann, and F. Noé, "DeepQMC: An open-source software suite for variational optimization of deep-learning molecular wave functions," J. Chem. Phys. **159**(9), 094108 (2023).

[23]T. A. R. Purcell, M. Scheffler, and L. M. Ghiringhelli, "Recent advances in the SISSO method and their implementation in the SISSO++ code," J. Chem. Phys. **159**(11), 114110 (2023).

[24]E. Gelžinytė, S. Wengert, T. K. Stenczel, H. H. Heenen, K. Reuter, G. Csányi, and N. Bernstein, "wfl Python toolkit for creating machine learning interatomic potentials and related atomistic simulation workflows," J. Chem. Phys. **159**(12), 124801 (2023).
[25]P. T. Salzbrenner, S. H. Joo, L. J. Conway, P. I. C. Cooke, B. Zhu, M. P. Matraszek, W. C. Witt, and C. J. Pickard, "Developments and further applications of ephemeral data derived potentials," J. Chem. Phys. **159**(14), 144801 (2023).
[26]J. A. Vita, E. G. Fuemmeler, A. Gupta, G. P. Wolfe, A. Q. Tao, R. S. Elliott, S. Martiniani, and E. B. Tadmor, "ColabFit exchange: Open-access datasets for data-driven interatomic potentials," J. Chem. Phys. **159**(15), 154802 (2023).
[27]W. C. Witt, C. van der Oord, E. Gelžinytė, T. Järvinen, A. Ross, J. P. Darby, C. H. Ho, W. J. Baldwin, M. Sachs, J. Kermode, N. Bernstein, G. Csányi, and C. Ortner, "ACEpotentials.jl: A Julia implementation of the atomic cluster expansion," J. Chem. Phys. **159**(16), 164101 (2023).
[28]M. F. Langer, J. T. Frank, and F. Knoop, "Stress and heat flux via automatic differentiation," J. Chem. Phys. **159**(17), 174105 (2023).
[29]S. Klawohn, J. P. Darby, J. R. Kermode, G. Csányi, M. A. Caro, and A. P. Bartók, "Gaussian approximation potentials: Theory, software implementation and application examples," J. Chem. Phys. **159**(17), 174108 (2023).
[30]J. Behler and M. Parrinello, "Generalized neural-network representation of high-dimensional potential-energy surfaces," Phys. Rev. Lett. **98**(14), 146401 (2007).
[31]A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, "Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons," Phys. Rev. Lett. **104**(13), 136403 (2010).