

From GAP to ACE to MACE

Commentary by

Noam Bernstein

Center for Materials Physics and Technology, U. S. Naval Research Laboratory, Washington, D.C.

on

1. **Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons**
A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, [Phys. Rev. Lett., 104:136403 \(2010\)](#)
2. **MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields**
I. Batatia, D. P. Kovács, G. Simm, C. Ortner, and G. Csányi, [Adv. Neural. Inf. Process. Syst., 35:11423 \(2022\)](#)

Received October 1, 2024

Published October 7, 2024

Statement of Significance

The Gaussian approximation potential (GAP) machine-learning-inspired functional form was the first to be used for a general-purpose interatomic potential. The atomic cluster expansion [1, 2] (ACE), previously the subject of a KIM Review [3], and its multilayer neural-network extension (MACE) have joined GAP among the methods widely used for machine-learning interatomic potentials. Here I review extensions to the original GAP formalism, as well as ACE and MACE-based frameworks that maintain the good features and mitigate the limitations of the original GAP approach.

Background: Gaussian Approximation Potentials with Smooth Overlap of Atomic Positions (SOAP-GAP)

Machine learning interatomic potentials (MLIPs) have revolutionized the field of atomistic simulations by replacing functional forms that are motivated by the specific physics of the bonding between atoms with minimally constrained many-body forms, as reviewed in [4–8]. Since any sufficiently flexible functional form could (by definition) reproduce an arbitrarily complicated potential energy surface (PES), one of the main practical challenges in developing MLIPs has been the tradeoff between this flexibility and the resulting tendency for inaccuracy away from the fitting data. If we had reference data available for all configuration (an infinite amount), that would fully constrain any fitting function. However, since generating reference data (e.g. with density functional theory calculations) is computationally expensive, reducing the amount of required data is essential. This can be achieved by taking advantage of features of the PES, such as symmetries or smoothness, that reduce the possible variation of the energy between fitting configurations, and hence the amount of data needed to constrain the fit. This is particularly important for the very flexible functions required to produce accurate and general purpose MLIPs that are applicable throughout a wide range of configurations and geometries.

About 15 years after the earliest proposals to use a machine-learning method, neural networks, for interatomic potentials [9], the first modern MLIPs succeeded in describing atomic geometries for specific physical systems [10, 11]. The Gaussian approximation potential (GAP) approach [11] uses sparse Gaussian process regression with a number of different descriptors of the atomic environments. One of the first efforts to develop a *general purpose* MLIP for a particular material was the silicon GAP [12] that used smooth overlap of atomic positions (SOAP) descriptors. This potential was able to reproduce the energies of its density functional theory (DFT) reference energies (and their force and stress gradients) for a single element in a wide range of crystal structures, defects (point, line, and plane), and disordered states such as liquid and amorphous. The SOAP-GAP functional form had exact rotation and permutation symmetry, as well as smoothness, due to the form of the squared exponential SOAP descriptors, and regularization (reducing the tendency to follow every fluctuation in the fitting data) through the “noise” hyperparameters of Gaussian process regression ¹. The flexibility of the function could be controlled through the SOAP descriptor parameters and the number of representative points in the sparse GPR.

This pioneering effort showed that an accurate general purpose MLIP could be developed, but the specific methods that made it possible also led to a number of limitations. The computational cost was large for an interatomic potential, of order 10-100 ms/atom on a single CPU core [13], and it scaled with the number of chemical elements squared. The use of GPR made the actual model fitting trivial by reducing it to the computation of the pseudo-inverse of the covariance matrix (via numerical linear algebra), but there was only mixed success in the initial attempts to use the GPR-predicted variance as a predictor of the MLIP error. Here I briefly survey some of the improvements and successors to SOAP-GAP that attempted to maintain the advantages and mitigate the disadvantages, while leveraging other mathematical and computational developments.

Improvements to SOAP-GAP

One change that minimally altered the overall SOAP-GAP form but led to substantial speedup was the development of new descriptors [14] implemented in the `soap_turbo` package [15]. The SOAP kernels represent the similarity of two atomic environments (i.e. the kernel for GPR) starting from a spherically symmetric Gaussian centered around each neighboring atom. In `soap_turbo`, these Gaussians are replaced with a product of a radial Gaussian and an angular Gaussian. This modification gives some additional flexibility, for example varying the smoothing, i.e. the width of the Gaussians, with neighbor distance, and crucially it is considerably faster to evaluate. With `soap_turbo` it is possible to match the accuracy of a conventional SOAP-GAP at $\sim 10\times$ lower computational cost [14]. As a result, these improved descriptors have essentially replaced the original SOAP variant in the QUIP `gap_fit` software [16].

A larger change, to address the quadratic scaling with the number of chemical elements, was proposed by Darby *et al.* through an approach called tensor-reduced density representations [17]. The SOAP-GAP expressions were re-cast in terms of quantities defined in the atomic cluster expansion (ACE) formalism [1, 2]. This redefinition makes it clear that the expansion of the SOAP descriptors into chemical elements and radial basis functions, whose number of tensor indices grows linearly with the number of chemical elements, can be transformed into an embedding ², i.e. a linear transformation into a space of arbitrary dimensionality, which can be converged to recover the exact tensor. The great advantage of this transformation is that the size of this space does not grow with combinatorial complexity as in earlier formulations. Darby *et al.* conclude that, in practice, the cost of the computation for evaluating the GAP becomes independent of the composition. They find that the tensor-reduced representation leads to a reduction by a factor of ~ 10 in the number of descriptor vector elements required for a given accuracy [17, 18]. Further exploration will be needed to determine the optimal size of the embedded space, the best method to optimize the linear transformation, and under what circumstances (e.g. number of elements) operating in the embedded space leads to a net reduction in computational cost.

GPR, which underlies GAP, predicts not only the value of the fit quantity, i.e. the potential energy, but also its variance, which is a measure of its uncertainty. It was initially hoped that this variance would be a

¹Note that most of this so-called noise is in fact model error due to the finite range of the potential. Actual uncertainty in the reference values, e.g. due to incomplete convergence, is much smaller.

²Here “embedding” is used in its machine-learning meaning, which is unrelated to the *emdedded* atom method (EAM) potentials.

good predictor of the MLIP’s error, and this was indeed demonstrated for the Si SOAP-GAP [12], as well as for a SOAP-GAP fit to energy barriers in heterogeneous catalysis [19]. However, in some other systems where SOAP-GAP was applied this was not the case, and we suspect that the successful examples depended on the availability of a large amount of fitting data relative to the diversity of configurations. Since the initial demonstration, a number of efforts have been made to improve GAP error prediction based on GPR-predicted variance. One approach applied to SOAP-GAP is to optimize the model’s hyperparameters³ by maximizing the Bayesian log-likelihood [20]. This is a computationally expensive process, but this cost is partially mitigated by a new parallelized implementation of the GAP fitting code [21]. Another approach that proved successful was to speed up hyperparameter optimization and avoid the need for sparsification of the representative points by limiting the amount of reference data and restricting the model to a specific system [22], or by limiting the model’s non-linearity (reducing its body order and potentially its accuracy) [23].

Beyond GAP

While the extensions to GAP discussed above have addressed some of its limitations, other approaches, with their own complementary advantages and disadvantages, are also under development. The two that I describe below, linear ACE as implemented in `ACEpotentials.jl` and nonlinear neural-network multilayer-ACE (MACE), were influenced by the same ideas and motivations as the design choices of SOAP-GAP.

Linear atomic cluster expansion via `ACEpotentials.jl`

As the work of Drautz [1–3] showed, a very wide range of interatomic potentials can be seen as aspects of a unified atomic cluster expansion (ACE) framework. These include SOAP-GAP, a non-linear model with 3-body descriptors, the spectral neighbor analysis potential (SNAP)[24, 25], a linear or quadratic model with 4-body descriptors, and a number of linear models with arbitrary body order more directly tied to the ACE functional form. Several codes implementing the latter approach are available, including moment tensor potentials (MTP) [26], performant ACE [27] and PACEMAKER [28, 29], and the implementation I discuss here. Combining the lessons of GAP with ACE led to the development of a new ACE fitting implementation, `ACEpotentials.jl` [30, 31]. As in SOAP-GAP, rotation and permutation symmetry are exact (due to the ACE framework), but smoothness and regularization are facilitated through entirely different means.

Rather than using the squared exponentials that smooth the atomic position variation in SOAP, `ACEpotentials.jl` uses a linear model with polynomial basis, whose oscillatory tendencies are quantified by the polynomial degree. A mathematical expression combining the radial and angular oscillations defines a *total* polynomial degree, whose maximum value is used to truncate the basis, with the possibility of separate control over radial and tangential variation. The fitting is done using Tikhonov or ridge regression, which adds a term to the minimized loss function related to the magnitude of the coefficients of each basis polynomial. This term is intended to make the function smooth by more strongly penalizing large coefficients, since they are likely to lead to unphysical oscillations away from the fitting data. Particular choices for the weight of each basis function’s coefficient in the ridge term can maximize particular quantifications of smoothness, such as the average curvature, or reproduce the smoothness of the SOAP squared-exponential kernels.

Ridge regression can also be analyzed through a Bayesian interpretation, where the ridge term can be thought of as a Bayesian prior favoring smooth fits. This formulation makes it possible to automatically optimize the level of regularization on a global (Bayesian ridge regression, BRR) or per-coefficient (automatic relevance determination, ARD) basis [30], and to produce an ensemble, or committee, of potentials for uncertainty quantification by drawing from the probability distribution of polynomial coefficients [32].

Like GAP, the scaling of `ACEpotentials.jl` potentials with the number of elements is a power law (with an exponent that depends on body order). Through the same analysis as for GAP, Darby *et al.* applied tensor

³Hyperparameters are parameters that control the behavior of a model but are set by the developer, either *a priori* or through some heuristics, in contrast to model parameters that are adjusted by the regression method to minimize the error on the fitting data.

reduction to ACE and showed similarly promising results, but these have not yet been explored in a wide range of applications, or incorporated into the current generally available implementation.

As its name suggests, the fitting framework of `ACEpotentials.jl` is implemented in Julia [33], and is mainly used in that environment. To use the resulting potentials, an atomic-simulation environment (ASE) calculator is available, and most, although not all, of the features can be exported into the format used by the performant ACE (PACE) implementation [27]. This high speed evaluation code can be used within the LAMMPS simulation software [34] for a wide range of time propagation and sampling methods with parallelism based on spatial domain decomposition.

Nonlinear neural-network multilayer atomic cluster expansion (MACE)

The ACE formulation was used as the basis for a number of multi-layer neural-network architectures, including NequIP [35] and multi-ACE [36]. Here I discuss a particular special case of the more general multi-ACE approach, MACE, which is a very flexible non-linear extension of the ACE framework [37, 38]. Its input is the ACE description of the atomic environment, starting from an atomic basis, forming a tensor product basis, and finally a rotationally invariant symmetrized basis [39]. Rather than using it as a basis for a linear model as in `ACEpotentials.jl`, this description is fed into an equivariant message-passing graph neural network (GNN) that couples messages, constructed from symmetrized-basis tensors, along edges between nodes corresponding to neighboring atoms. Each layer constructs output node features based on input messages from neighboring atom nodes, and the final, readout, layer predicts output energies (or, in principle, other tensor quantities) based on the final node features. As with all neural networks, there is the potential for extreme flexibility due to the range of options for each of these stages.

In practice, some simple choices for the details of the architecture have led to remarkably accurate and stable MLIPs across a vast range of materials. In general the tensor product basis is truncated to body order 4, two GNN layers are used, and the messages include angular momentum components of at most 2. These choices limit the nonlinearity to the tensor product basis construction, and to the radial basis functions and energy readout function, which are implemented as fully connected NNs (multilayer perceptrons, MLPs). This mild degree of nonlinearity is meant to strike an optimal balance between functional form flexibility on the one hand, and ease of fitting on the other.

The most remarkable result is that, in addition to making more accurate material-specific MLIPs than GAP and `ACEpotentials.jl` [19, 39, 40], MACE has demonstrated the ability to create a universal foundation model that is applicable across essentially the entire periodic table. MACE-MP0 [41], based on the Materials Project (MP) database [42, 43], and MACE-OFF23 [44], based mainly on the SPICE database [45], have demonstrated accurate results and stable dynamics and sampling across extremely wide ranges of configuration space. MACE-MP0 achieves good accuracy across 89 elements in structures represented in the MP project (about 20 meV/atom energy MAE and 45 meV/Å force MAE for the medium model), and at least qualitatively reasonable description for 30 tested applications far beyond the configurations present in the fitting data. While the precise properties it predicts are often approximate, it is nearly always stable enough to carry out dynamics that can be used to efficiently generate new fitting configurations for fine-tuning. MACE-OFF23, fit to organic molecules containing 10 elements, is even more accurate (about 1-2 meV/atom energy RMSE and 20-30 meV/Å force RMSE), although unlike MACE-MP0 its fitting and testing data consist almost entirely of equilibrium configurations.

The MACE potential is currently implemented [38] using the `pytorch` library [46], with a python script for fitting, an ASE Calculator interface for evaluation from python, and a `torchscript`-based interface for use in LAMMPS [34].

Direct comparison

To give a more concrete idea of the relative accuracy and speed of these MLIPs, I fit GAP, ACE, and MACE models to a database of $\text{Cu}_x\text{Al}_{1-x}$ DFT calculations. Note that the comparison below is meant to give an

Table 1: Root-mean-squared error for fitting and testing data on energy (meV/atom), forces (meV/Å), and stress (meV/Å³), and computational time in ms/atom/time step for MD of a 1024 atom configuration on a single Intel Xeon Gold 6226R 2.9 GHz CPU core (GAP, ACE) or NVIDIA A100 GPU (MACE).

potential	fit error			test error			time
	energy	force	stress	energy	force	stress	
GAP	93	80	3.3	8.2	78	3.0	2.1
ACE 3, 10	140	140	6.1	18	140	5.3	0.057
ACE 5, 12	100	84	2.4	8.5	82	2.4	0.18
MACE small	95	86	1.6	5.8	76	1.7	0.042
MACE medium	96	82	1.4	5.7	80	1.7	0.12

overall sense of their relative performance, with the caveat that these results are based on somewhat arbitrary heuristics for the various hyperparameters, and could vary significantly for other choices.

The fitting set consists of 4587 configurations from an iterative random-structure search and GAP fitting process [47, 48] supplemented with automatically generated supercells, including various defects, at a range of temperatures up to and including the liquid phase. An additional 682 configurations from the later iterations of this process were reserved for testing. Reference energies, forces, and stresses were computed using DFT-PBE [49] with the VASP software [50]. A GAP model was fit using the default heuristics of `gap_rss_iter_fit` [48], with two SOAP descriptors per element, and a total of 4000 sparse points. Two ACE models were fit using the GAP cutoff of 6.5 Å: a faster potential with body order 3 and maximum polynomial degree 10 (338 basis functions), and a slower potential with body order 5 and polynomial degree 12 (5370 basis functions). Finally, two MACE models were fit by fine-tuning (with multihead stabilization) the small and medium foundation models, which differ in hyperparameters affecting their speed, memory use, and accuracy, from Ref. [41]. These notably differ from GAP and ACE in their effective interaction range of 12 Å.

In Table 1, I show the error on the fitting and testing databases and the computational cost of the potential on a single CPU core (GAP, ACE) or GPU (MACE) in a 1024 atom molecular-dynamics (MD) simulation using LAMMPS [34]. This system size is chosen to be large enough that the GPU-related latency is a small contribution and the listed times will be roughly constant for larger systems. All three MLIPs produce accurate models, with errors that, in our experience, correspond to useful accuracy (of order 5-10%) for physical properties such as binding energies, elastic constants, defect formation energies, etc. The faster ACE model is considerably less accurate than GAP, while the slower one matches GAP in accuracy. Both MACE models have very similar accuracy, matching or exceeding that of the GAP and ACE models. This may be due to the nonlinearity and additional parameters in the MACE functional form, as well as the longer cutoff.

The computational speed is somewhat harder to compare, because GAP and ACE scale nearly perfectly linearly in the number of atoms and their current implementations are efficiently parallelized using domain decomposition, while MACE has a significant latency due to CPU-GPU communication and has not yet been efficiently parallelized. The table comparison is in the regime of a large number of atoms (~ 1000) per CPU core (GAP, ACE) or GPU (MACE). GAP is the slowest of the MLIPs, at about 2 ms/atom/time step. The two ACE variants show some of the available tradeoff range, with the less accurate one $\sim 35\times$ faster than GAP and the more accurate one only $\sim 12\times$. The two MACE potentials have a comparable speed on a single GPU to ACE on a single CPU core.

The constraints of the current MACE implementation to a single GPU mean that the largest system size on a GPU with 80 GB of memory is about 21,000 for the small model and 8,000 atoms for the medium model. With the small model, the largest system can be simulated for $\sim 0.11 \times 10^6$ time-steps/day, but due to latency even a 54 atom system can only be simulated for $\sim 8 \times 10^6$ time-steps/day, about five times slower per atom. ACE, on the other hand, is comparable in per-atom speed on a single CPU core to MACE on a GPU, and when run in parallel loses only about 30% in speed with only 100 atoms/CPU core. Therefore, a single 128 core node should be able to simulate 21,000 atoms with the faster, less accurate ACE for $\sim 7 \times 10^6$ time steps/day, about $60\times$ faster than MACE. The development of efficient CPU implementations of MACE, as well as improvements

to its domain decomposition parallel efficiency for both the GPU and CPU versions, are ongoing.

Outlook

Rapid advances in MLIPs are beginning to remake the field of atomistic simulations of material properties. The success of SOAP-GAP showed that it was possible to develop sufficiently expressive functional forms to describe a material's PES across a wide range of geometries, but with sufficient smoothness that they could be fit to a practical amount of DFT-calculated data. New advances in descriptors for GAP will make it possible to apply this capability to larger and more chemically complex systems.

ACE models can match the accuracy of GAP models, but are one to two orders of magnitude faster, even when compared to the relatively fast `soap_turbo` formulation. This speedup can enable the simulation of larger systems, for example approaching the scale of multiple crystalline grains separated by grain boundaries (of order $10^4 - 10^5$ atoms), or enable longer time sampling (of order nanoseconds, 10^6 time steps).

MACE enables the creation of foundation models that describe nearly all chemical elements and can be used as a basis for fine-tuned models for a specific material system or reference data calculation method. This capability has already revolutionized approaches to simulating new, chemically complex systems, including systems as diverse as solid state acid proton conductors, polymer-based Li-ion sulfide electrolytes, and structural metallic alloys.

Especially by combining the best ingredients of GAP, ACE, and MACE, as well as the many others I did not have time or space to present, MLIPs are becoming capable of dramatically more accurate calculations of properties for systems with chemical and structural complexity that approaches experimentally and technologically relevant materials.

References

- 1 R. Drautz, "Atomic cluster expansion for accurate and transferable interatomic potentials", *Phys. Rev. B*, 99(1):014104, 2019, <https://link.aps.org/doi/10.1103/PhysRevB.99.014104>.
- 2 R. Drautz, "Erratum: Atomic cluster expansion for accurate and transferable interatomic potentials [Phys. Rev. B 99, 014104 (2019)]", *Phys. Rev. B*, 100(24):249901, 2019, <https://link.aps.org/doi/10.1103/PhysRevB.100.249901>.
- 3 C. Ortner, "On the Atomic Cluster Expansion: interatomic potentials and beyond", *KIMREVIEW*, 1(1):1–5, 2023, <https://kimreview.org/commentaries/10-25950-c7f24234/>.
- 4 A. P. Bartók, S. De, C. Poelking, N. Bernstein, J. R. Kermode, G. Csányi, and M. Ceriotti, "Machine learning unifies the modeling of materials and molecules", *Science Advances*, 3(12):e1701816, 2017, <http://advances.sciencemag.org/lookup/doi/10.1126/sciadv.1701816>.
- 5 V. L. Deringer, M. A. Caro, and G. Csányi, "Machine Learning Interatomic Potentials as Emerging Tools for Materials Science", *Advanced Materials*, 31(46):1902765, 2019, <https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.201902765>.
- 6 J. Behler and G. Csányi, "Machine learning potentials for extended systems: a perspective", *Eur. Phys. J. B*, 94(7):142, 2021, <https://doi.org/10.1140/epjb/s10051-021-00156-1>.
- 7 V. L. Deringer, A. P. Bartók, N. Bernstein, D. M. Wilkins, M. Ceriotti, and G. Csányi, "Gaussian Process Regression for Materials and Molecules", *Chem. Rev.*, 121(16):10073–10141, 2021, <https://doi.org/10.1021/acs.chemrev.1c00022>.
- 8 F. Musil, A. Grisafi, A. P. Bartók, C. Ortner, G. Csányi, and M. Ceriotti, "Physics-Inspired Structural Representations for Molecules and Materials", *Chem. Rev.*, 121(16):9759–9815, 2021, <https://doi.org/10.1021/acs.chemrev.1c00021>.

- 9 A. J. Skinner and J. Q. Broughton, "Neural Networks in Computational Materials Science - Training Algorithms", *Model Simul Mater Sc*, 3(3):371–390, 1995, <http://stacks.iop.org/0965-0393/3/i=3/a=006?key=crossref.bc959a676e227f670213d87bacea379e>.
- 10 J. Behler and M. Parrinello, "Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces", *Phys. Rev. Lett.*, 98(14):146401, 2007, <http://link.aps.org/doi/10.1103/PhysRevLett.98.146401>.
- 11 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, <https://link.aps.org/doi/10.1103/PhysRevLett.104.136403>.
- 12 A. P. Bartók, J. Kermode, N. Bernstein, and G. Csányi, "Machine Learning a General-Purpose Interatomic Potential for Silicon", *Phys. Rev. X*, 8(4):1939, 2018, <https://link.aps.org/doi/10.1103/PhysRevX.8.041048>.
- 13 Y. Zuo, C. Chen, X. Li, Z. Deng, Y. Chen, J. Behler, G. Csányi, A. V. Shapeev, A. P. Thompson, M. A. Wood, and S. P. Ong, "Performance and Cost Assessment of Machine Learning Interatomic Potentials", *J. Phys. Chem. A*, 124(4):731–745, 2020, <https://doi.org/10.1021/acs.jpca.9b08723>.
- 14 M. A. Caro, "Optimizing many-body atomic descriptors for enhanced computational performance of machine learning based interatomic potentials", *Phys. Rev. B*, 100(2):024112, 2019, <https://link.aps.org/doi/10.1103/PhysRevB.100.024112>.
- 15 M. A. Caro, *soap_turbo*, https://github.com/libAtoms/soap_turbo (visited on 08/08/2024).
- 16 G. Csányi, N. Bernstein, and J. R. Kermode, *QUIP*, <https://github.com/libAtoms/QUIP> (visited on 08/08/2024).
- 17 J. P. Darby, D. P. Kovács, I. Batatia, M. A. Caro, G. L. W. Hart, C. Ortner, and G. Csányi, "Tensor-Reduced Atomic Density Representations", *Phys. Rev. Lett.*, 131(2):028001, 2023, <https://link.aps.org/doi/10.1103/PhysRevLett.131.028001>.
- 18 L. Shenoy, C. D. Woodgate, J. B. Staunton, A. P. Bartók, C. S. Becquart, C. Domain, and J. R. Kermode, "Collinear-spin machine learned interatomic potential for Fe₇Cr₂Ni alloy", *Phys. Rev. Mater.*, 8(3):033804, 2024, <https://link.aps.org/doi/10.1103/PhysRevMaterials.8.033804>.
- 19 L. L. Schaaf, E. Fako, S. De, A. Schäfer, and G. Csányi, "Accurate energy barriers for catalytic reaction pathways: an automatic training protocol for machine learning force fields", *npj Comput Mater*, 9(1):1–10, 2023, <https://www.nature.com/articles/s41524-023-01124-2>.
- 20 A. P. Bartók and J. R. Kermode, *Improved uncertainty quantification for Gaussian process regression based interatomic potentials*, <http://arxiv.org/abs/2206.08744> (visited on 08/26/2024).
- 21 S. Klawohn, J. R. Kermode, and A. P. Bartók, "Massively parallel fitting of Gaussian approximation potentials", *Mach. Learn.: Sci. Technol.*, 4(1):015020, 2023, <https://dx.doi.org/10.1088/2632-2153/aca743>.
- 22 R. Jinnouchi, F. Karsai, and G. Kresse, "On-the-fly machine learning force field generation: Application to melting points", *Phys. Rev. B*, 100(1):014105, 2019, <https://link.aps.org/doi/10.1103/PhysRevB.100.014105>.
- 23 J. Vandermause, S. B. Torrisi, S. Batzner, Y. Xie, L. Sun, A. M. Kolpak, and B. Kozinsky, "On-the-fly active learning of interpretable Bayesian force fields for atomistic rare events", *npj Comput Mater*, 6(1):1–11, 2020, <https://www.nature.com/articles/s41524-020-0283-z>.
- 24 A. P. Thompson, L. P. Swiler, C. R. Trott, S. M. Foiles, and G. J. Tucker, "Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials", *Journal of Computational Physics*, 285:316–330, 2015, <https://www.sciencedirect.com/science/article/pii/S0021999114008353>.
- 25 M. A. Wood and A. P. Thompson, "Extending the accuracy of the SNAP interatomic potential form", *The Journal of Chemical Physics*, 148(24):241721, 2018, <https://doi.org/10.1063/1.5017641>.

- 26 A. V. Shapeev, "Moment Tensor Potentials: A Class of Systematically Improvable Interatomic Potentials", *Multiscale Model Sim*, 14(3):1153–1173, 2016, <http://epubs.siam.org/doi/10.1137/15M1054183>.
- 27 Y. Lysogorskiy, C. v. d. Oord, A. Bochkarev, S. Menon, M. Rinaldi, T. Hammerschmidt, M. Mrovec, A. Thompson, G. Csányi, C. Ortner, and R. Drautz, "Performant implementation of the atomic cluster expansion (PACE) and application to copper and silicon", *npj Comput Mater*, 7(1):1–12, 2021, <https://www.nature.com/articles/s41524-021-00559-9>.
- 28 A. Bochkarev, Y. Lysogorskiy, S. Menon, M. Qamar, M. Mrovec, and R. Drautz, "Efficient parametrization of the atomic cluster expansion", *Phys. Rev. Mater.*, 6(1):013804, 2022, <https://link.aps.org/doi/10.1103/PhysRevMaterials.6.013804>.
- 29 *pacemaker*, <https://github.com/ICAMS/python-ace> (visited on 08/28/2024).
- 30 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", *The Journal of Chemical Physics*, 159(16):164101, 2023, <https://doi.org/10.1063/5.0158783>.
- 31 *ACEpotentials.jl*, <https://github.com/ACEsuit/ACEpotentials.jl> (visited on 08/08/2024).
- 32 C. van der Oord, M. Sachs, D. P. Kovács, C. Ortner, and G. Csányi, "Hyperactive learning for data-driven interatomic potentials", *npj Comput Mater*, 9(1):1–14, 2023, <https://www.nature.com/articles/s41524-023-01104-6>.
- 33 J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A Fresh Approach to Numerical Computing", *SIAM Rev.*, 59(1):65–98, 2017, <https://epubs.siam.org/doi/10.1137/141000671>.
- 34 A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton, "LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales", *Computer Physics Communications*, 271:108171, 2022, <https://www.sciencedirect.com/science/article/pii/S0010465521002836>.
- 35 S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, "E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials", *Nat Commun*, 13(1):2453, 2022, <https://www.nature.com/articles/s41467-022-29939-5>.
- 36 A. Bochkarev, Y. Lysogorskiy, C. Ortner, G. Csányi, and R. Drautz, "Multilayer atomic cluster expansion for semilocal interactions", *Phys. Rev. Res.*, 4(4):L042019, 2022, <https://link.aps.org/doi/10.1103/PhysRevResearch.4.L042019>.
- 37 I. Batatia, D. P. Kovács, G. Simm, C. Ortner, and G. Csányi, "MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields", *Advances in Neural Information Processing Systems*, 35:11423–11436, 2022, https://proceedings.neurips.cc/paper_files/paper/2022/hash/4a36c3c51af11ed9f34615b81edb5bbc-Abstract-Conference.html.
- 38 *mace*, <https://github.com/ACEsuit/mace> (visited on 08/22/2024).
- 39 D. P. Kovács, I. Batatia, E. S. Arany, and G. Csányi, "Evaluation of the MACE force field architecture: From medicinal chemistry to materials science", *The Journal of Chemical Physics*, 159(4):044118, 2023, <https://doi.org/10.1063/5.0155322>.
- 40 W. J. Baldwin, X. Liang, J. Klarbring, M. Dubajic, D. Dell'Angelo, C. Sutton, C. Caddeo, S. D. Stranks, A. Mattoni, A. Walsh, and G. Csányi, "Dynamic Local Structure in Caesium Lead Iodide: Spatial Correlation and Transient Domains", *Small*, 20(3):2303565, 2024, <https://onlinelibrary.wiley.com/doi/abs/10.1002/smll.202303565>.
- 41 I. Batatia et al., *A foundation model for atomistic materials chemistry*, <http://arxiv.org/abs/2401.0096> (visited on 08/08/2024).

- 42 A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, “Commentary: The Materials Project: A materials genome approach to accelerating materials innovation”, *APL Materials*, 1(1):011002, 2013, <https://doi.org/10.1063/1.4812323>.
- 43 *Materials Project*, <https://next-gen.materialsproject.org/> (visited on 08/08/2024).
- 44 D. P. Kovács, J. H. Moore, N. J. Browning, I. Batatia, J. T. Horton, V. Kapil, W. C. Witt, I.-B. Magdău, D. J. Cole, and G. Csányi, *MACE-OFF23: Transferable Machine Learning Force Fields for Organic Molecules*, <http://arxiv.org/abs/2312.15211> (visited on 08/08/2024).
- 45 P. Eastman, P. K. Behara, D. L. Dotson, R. Galvelis, J. E. Herr, J. T. Horton, Y. Mao, J. D. Chodera, B. P. Pritchard, Y. Wang, G. De Fabritiis, and T. E. Markland, “SPICE, A Dataset of Drug-like Molecules and Peptides for Training Machine Learning Potentials”, *Sci Data*, 10(1):11, 2023, <https://www.nature.com/articles/s41597-022-01882-6>.
- 46 J. Ansel, E. Yang, H. He, N. Gimelshein, A. Jain, M. Voznesensky, B. Bao, P. Bell, D. Berard, E. Burovski, G. Chauhan, A. Chourdia, W. Constable, A. Desmaison, Z. DeVito, E. Ellison, W. Feng, J. Gong, M. Gschwind, B. Hirsh, S. Huang, K. Kalambarkar, L. Kirsch, M. Lazos, M. Lezcano, Y. Liang, J. Liang, Y. Lu, C. K. Luk, B. Maher, Y. Pan, C. Puhersch, M. Reso, M. Saroufim, M. Y. Siraichi, H. Suk, S. Zhang, M. Suo, P. Tillet, X. Zhao, E. Wang, K. Zhou, R. Zou, X. Wang, A. Mathews, W. Wen, G. Chanan, P. Wu, and S. Chintala, “PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation”, *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, vol. 2, ASPLOS '24, New York, NY, USA: Association for Computing Machinery, 2024, 929–947, <https://dl.acm.org/doi/10.1145/3620665.3640366> (visited on 08/22/2024).
- 47 N. Bernstein, G. Csányi, and V. L. Deringer, “De novo exploration and self-guided learning of potential-energy surfaces”, *npj Computational Materials* 2018 4:1, 5(1):1–9, 2019, <https://www.nature.com/articles/s41524-019-0236-6>.
- 48 N. Bernstein, *How to do GAP-RSS — workflow documentation*, <https://libatoms.github.io/workflow/workflows.rss.html> (visited on 08/22/2024).
- 49 J. P. Perdew, K. Burke, and M. Ernzerhof, “Generalized Gradient Approximation Made Simple”, *Phys. Rev. Lett.*, 77(18):3865–3868, 1996, <https://link.aps.org/doi/10.1103/PhysRevLett.77.3865>.
- 50 G. Kresse and J. Furthmüller, “Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set”, *Phys. Rev. B*, 54(16):11169–11186, 1996, <https://link.aps.org/doi/10.1103/PhysRevB.54.11169>.