
Learning Interatomic Potentials at Multiple Scales

Xiang Fu
MIT CSAIL
xiangfu@mit.edu

Albert Musaelian
Harvard John A. Paulson
School of Engineering and Applied Sciences
albym@g.harvard.edu

Anders Johansson
Harvard John A. Paulson
School of Engineering and Applied Sciences
andersjohansson@g.harvard.edu

Tommi Jaakkola
MIT CSAIL
tommi@csail.mit.edu

Boris Kozinsky
Harvard John A. Paulson School of Engineering and Applied Sciences and
Robert Bosch Research and Technology Center
bkoz@g.harvard.edu

Abstract

The need to use a short time step is a key limit on the speed of molecular dynamics (MD) simulations. Simulations governed by classical potentials are often accelerated by using a multiple-time-step (MTS) integrator that evaluates certain potential energy terms that vary more slowly than others less frequently. This approach is enabled by the simple but limiting analytic forms of classical potentials. Machine learning interatomic potentials (MLIPs), in particular recent equivariant neural networks, are much more broadly applicable than classical potentials and can faithfully reproduce the expensive but accurate reference electronic structure calculations used to train them. They still, however, require the use of a single short time step, as they lack the inherent term-by-term scale separation of classical potentials. This work introduces a method to learn a scale separation in complex interatomic interactions by co-training two MLIPs. Initially, a small and efficient model is trained to reproduce short-time-scale interactions. Subsequently, a large and expressive model is trained jointly to capture the remaining interactions not captured by the small model. When running MD, the MTS integrator then evaluates the smaller model for every time step and the larger model less frequently, accelerating simulation. Compared to a conventionally trained MLIP, our approach can achieve a significant speedup ($\sim 3x$ in our experiments) without a loss of accuracy on the potential energy or simulation-derived quantities.

1 Introduction

The interatomic potential energy that governs the dynamics of a system of atoms has long been both understood and modeled as a combination of atomic interactions of various strengths and scales. In a system containing a comparatively stiff molecule in a soft fluid, for example, the intramolecular forces are much stronger than the intermolecular forces from the solvent. Classical potentials, such as the popular optimized potentials for liquid simulations (OPLS, [1, 2]), explicitly define the potential as such a sum over simple analytic terms:

$$E(\mathbf{r}) = E_{\text{bonds}}(\mathbf{r}) + E_{\text{angles}}(\mathbf{r}) + E_{\text{dihedrals}}(\mathbf{r}) + E_{\text{nb}}(\mathbf{r})$$

where \mathbf{r} denotes the atom positions; $E_{\text{bonds}}(\mathbf{r})$, $E_{\text{angles}}(\mathbf{r})$, and $E_{\text{dihedrals}}(\mathbf{r})$ are intramolecular (or “bonded”) bond, angle, and torsional potentials; and E_{nb} denotes the nonbonded, including intermolecular potential term. Rigid interactions such as bond vibrations, governed by E_{bonds} , occur at fast time scales, while the nonbonded interactions E_{nb} are slower and smoother. Due to the greater number of atoms involved in nonbonded interactions, however, the computational expense of calculating the E_{nb} term can be meaningfully larger.

Essentially, the intramolecular and intermolecular nonbonded forces are of different scales. While the intramolecular forces require a short time step, integrating the intermolecular forces at the same step size is often overkill. To harness this separation of both scales and computational cost, multiple-time-step (MTS) integrators [3–7] integrate the fast-evolving terms with a short time step and the slow-evolving terms with a long time step. MTS integrators are theoretically principled and have been widely used in various classical MD workflows [8–12], usually bringing a speedup of two to four times.

The approximations and limited functional forms of classical potentials are not sufficient for many applications. Ab initio molecular dynamics (AIMD) simulations—governed by a potential energy surface computed with electronic structure methods such as Density Functional Theory (DFT)—are widely used but suffer from dramatic computational limitations on time- and length-scale due to the expense and unfavorable scaling of the electronic structure calculations. The application of MTS schemes to AIMD has been limited by the difficulties in decomposing the ab initio potential into components of separate scales in the absence of a simple analytical form [13, 14].

Machine learning interatomic potentials (MLIPs) [15–48] are increasingly used to run MD simulations orders of magnitude cheaper than AIMD while preserving near-AIMD accuracy. MLIPs also largely avoid classical potentials’ assumption of an explicit discrete covalent bond topology, which greatly broadens their applicability, notably in materials science and reactive simulations. While MLIPs have enabled many previously impractical or impossible simulations, further improvements in the speed and cost of MLIP-driven MD are extremely valuable. MTS methods remain largely unexplored for MLIPs: although recent research [49] has combined an infrequently evaluated MLIP with a classical potential evaluated at every time step, a methodology for machine learning a scale-separated potential model from data is, to the best of our knowledge, lacking in the literature. This work presents an approach for learning complementary scale-separated MLIPs from data, yielding a $\sim 3\times$ speedup in MD simulations using MTS integration.

2 Preliminaries

MD simulation involves integrating a Newtonian equation of motion $\ddot{\mathbf{r}} = d^2\mathbf{r}/dt^2 = \mathbf{m}^{-1}\mathbf{F}(\mathbf{r})$ with atomic positions \mathbf{r} , masses \mathbf{m} , and forces \mathbf{F} . The forces are obtained by differentiating the potential energy of a molecular system $E(\mathbf{r})$ with respect to the atomic positions \mathbf{r} : $\mathbf{F}(\mathbf{r}) = -\partial E/\partial \mathbf{r}$.

Allegro [37] is an $E(3)$ -equivariant [50] neural network MLIP architecture for predicting $E(\mathbf{r})$ that enforces strictly local interactions while achieving state-of-the-art performance. Its strict locality enables efficient parallel scaling across GPUs to reach larger length- and time-scales [51]. In Allegro, every pair of atoms within a chosen fixed cutoff distance r are considered neighbors. For each ordered pair of neighboring atoms i and j , an Allegro model produces a learnable, $E(3)$ -invariant, many-body final latent representation \mathbf{x}^{ij} of the geometry of i , j , and all other neighbors of i . An edge energy E_{ij} is then predicted from it via the output block, a multi-layer perception MLP_{out} without bias; the sum over edge energies gives the total potential energy:

$$E_{\text{ML}}(\mathbf{r}) = \sum_{(i,j):\|\mathbf{r}_i-\mathbf{r}_j\|\leq r} E_{ij} = \sum_{(i,j):\|\mathbf{r}_i-\mathbf{r}_j\|\leq r} \text{MLP}_{\text{out}}(\mathbf{x}^{ij}) \quad (1)$$

An Allegro model’s predicted $E_{\text{ML}}(\mathbf{r})$ is trained to reproduce the potential energy and forces from a reference method such as DFT using a loss function like:

$$\mathcal{L} = \lambda_E \left\| \frac{E_{\text{ML}} - E}{N} \right\|^2 + \lambda_F \frac{1}{3N} \left\| -\frac{\partial E_{\text{ML}}}{\partial \mathbf{r}} - \mathbf{F} \right\|^2 \quad (2)$$

where λ_E and λ_F are loss coefficients for energy and forces and N is the number of atoms. For complete details on Allegro and its training see [37].

MTS Integrators accelerate MD simulations by propagating different parts of the dynamics with different time steps suited to their respective characteristic time scales. In this paper, we focus on the reversible reference system propagator algorithms (rRESPA [5, 52]), an MTS integrator popular for its rigorous derivation, time-reversibility, and symplectic properties. We show the rRESPA algorithm in [Algorithm 1](#) and refer interested readers to [5] for detailed derivations.

Algorithm 1 An integration step of the MTS integrator (rRESPA)

```

1: Input: Inner time step  $\Delta t$ , number of inner time steps per outer time step  $N_{\text{inner}}$ , short-range force  $\mathbf{F}_s$ , long-range force  $\mathbf{F}_l$ , atom masses  $\mathbf{m}$ , initial positions  $\mathbf{r}$ , initial velocities  $\dot{\mathbf{r}}$ 
2:  $\dot{\mathbf{r}} \leftarrow \dot{\mathbf{r}} + \frac{1}{2}(N_{\text{inner}}\Delta t) \cdot \mathbf{m}^{-1}\mathbf{F}_l(\mathbf{r})$ 
3: for step  $i = 1, \dots, N_{\text{inner}}$  do
4:    $\dot{\mathbf{r}} \leftarrow \dot{\mathbf{r}} + \frac{1}{2}\Delta t \cdot \mathbf{m}^{-1}\mathbf{F}_s(\mathbf{r})$ 
5:    $\mathbf{r} \leftarrow \mathbf{r} + \dot{\mathbf{r}}\Delta t$ 
6:    $\dot{\mathbf{r}} \leftarrow \dot{\mathbf{r}} + \frac{1}{2}\Delta t \cdot \mathbf{m}^{-1}\mathbf{F}_s(\mathbf{r})$ 
7: end for
8:  $\dot{\mathbf{r}} \leftarrow \dot{\mathbf{r}} + \frac{1}{2}(N_{\text{inner}}\Delta t) \cdot \mathbf{m}^{-1}\mathbf{F}_l(\mathbf{r})$ 

```

3 Learning Scale Separation

To achieve scale separation and harness the MTS integrator, we need to enable efficient calculation of stiff and fast-evolving force terms. Meanwhile, we still need to capture the smooth and slow-evolving force terms so that the overall machine learning potential remains accurate. For many molecular systems of interest, short-range interactions, such as covalent bonds, are strong and induce stiff motions, while long-range interactions, such as non-bonded interactions, can be integrated with a longer time step.

The above observation motivates separating scales by combining MLIPs with different receptive fields. Allegro’s unique combination of the leading accuracy of equivariant techniques with strict locality particularly lends itself to such a scale separation scheme. We train two models:

- **Inner model** (E_{inner}): An efficient model with fewer parameters and interaction layers and a small radial cutoff that captures short-time-scale interactions.
- **Outer model** (E_{outer}): An expressive model with a larger number of parameters and interaction layers and a larger radial cutoff that fits the remaining interactions not learned by the inner model.

We let the two models jointly predict the potential energy: $E_{\text{ML}}(\mathbf{r}) = E_{\text{inner}}(\mathbf{r}) + E_{\text{outer}}(\mathbf{r})$. Training the two models together from scratch, however, may not induce scale separation: the outer model has sufficient capacity to learn the entire reference potential energy surface, and so the short-range interactions may not be attributed to the inner model. To avoid such degeneracy, at the beginning of training, we first freeze all parameters in the outer model to let the inner model fit the force and energy within its capacity. We then later start training the outer model with a zero-initialization over its output block MLP_{out} . This zero initialization ensures $E_{ij}^{\text{outer}} = 0$ and $\partial E_{ij}^{\text{outer}} / \partial \mathbf{r} = 0$ for all atom pairs (i, j) , preventing the initial noise of the outer model from interfering with the interactions already learned by the inner model. Zero-initialization has been widely used in previous works for finetuning pretrained models [53]. The training procedure is presented in [Algorithm 2](#). We refer to our scale-separated Allegro model as **MTS-Allegro**.

4 Experiments

Our experiments consider an ab initio water system [54]. This dataset contains 1593 reference calculations of bulk liquid water at the revPBE0-D3 level of accuracy. Each structure contains 192 atoms (64 water molecules). We randomly sample 1000 structures for training, 100 structures for validation, and the rest for testing¹. Both the Allegro model and the outer model of MTS-Allegro have

¹These structures are not sampled under an equilibrium condition such as NVE or NVT ensembles.

Algorithm 2 Learning MLIPs at multiple scales

- 1: **Input:** Inner model E_{inner} and outer model E_{outer} , inner training number of epoch M_{inner} , total training number of epoch M
 - 2: **Output:** Trained models E_{inner} and E_{outer}
 - 3: Zero-initialize the outer model’s output layer: MLP_{out} of E_{outer}
 - 4: **for** epoch $i = 1, \dots, M_{\text{inner}}$ **do**
 - 5: train $E_{\text{ML}} = E_{\text{inner}}$ through the loss defined in Equation (2)
 - 6: **end for**
 - 7: **for** epoch $i = M_{\text{inner}} + 1, \dots, M$ **do**
 - 8: train $E_{\text{ML}} = E_{\text{inner}} + E_{\text{outer}}$ through the loss defined in Equation (2)
 - 9: **end for**
-

Table 1: Energy and Force prediction mean absolute error (MAE) and root mean square error (RMSE).

	Allegro	MTS-Allegro	MTS-Allegro, inner
Energy MAE [meV/Atom]	2.4	1.6	12.5
Forces MAE [meV/Å]	40.3	35.5	72.5
Forces RMSE [meV/Å]	77.4	76.9	118.3

two interaction layers, a 6 Å cutoff, and the same parameter count; the inner model of MTS-Allegro has one interaction layer, a 4 Å cutoff, and half the width in each layer compared to the outer model. Detailed hyperparameters are included in [Appendix A](#).

We compare the accuracy in recovering the reference potential and various simulation-derived quantities between (1) our proposed MTS-Allegro model with rRESPA integration, (2) a conventionally trained Allegro model with standard Velocity Verlet integration, and (3) the inner model of MTS-Allegro alone with Velocity Verlet integration. We use a time step of 0.5 fs for Velocity Verlet integration and the inner loop of MTS integration, which is standard for water simulations [55–57]. For MTS-Allegro, we experiment with outer time steps of [1.0, 2.0, 3.0, 4.0] fs, corresponding to [2x, 4x, 6x, 8x] multiples of the inner time step.

Force and energy prediction accuracy. We report the force and energy prediction errors in [Table 1](#). The MTS-Allegro model has a very similar performance to a conventionally trained Allegro model. Unsurprisingly, the inner model of MTS-Allegro alone obtains higher errors due to its limited receptive field and capacity.

Structure and dynamics. To investigate MTS-Allegro’s ability to reproduce structural and dynamical observables, we simulate the water system in a canonical (NVT) ensemble (constant volume and temperature) at 300 K and compute the element-wise radial distribution functions (RDFs, structural) and the mean squared displacement (MSD, dynamical). We simulate for 400 picoseconds (ps) and remove the first 50 ps for equilibration when computing the observables. All models remain stable throughout the entire simulation. [Figure 1](#) (a-d) shows the RDFs and MSD (in log-log space) for Allegro, the inner model of MTS-Allegro, and MTS-Allegro under different outer step sizes. MTS-Allegro simulations with 2x to 8x outer time steps all achieve excellent agreement with Allegro on the RDFs and MSD, while the inner model alone produces erroneous dynamics and observables due to its limited capacity and, thereby, accuracy in recovering the potential.

Energy conservation under the microcanonical ensemble. For a correct MD simulation in the microcanonical (NVE) ensemble, the total energy (sum of potential and kinetic energies) should remain constant over time. To evaluate the energy conservation property of the MLIPs, we initialize the water system using a structure from the test dataset and a temperature of 300 K, run energy minimization, and then simulate for 100 ps in the NVE ensemble. [Figure 1](#) (e) shows the drift of total energy from the first frame (after removing the first 20 ps for equilibration). We observe that Allegro and MTS-Allegro with 2x and 4x outer time steps are energy-conserving. MTS-Allegro with 6x and 8x outer time steps yield drifts of 0.12 meV/(atom·ps) and 0.18 meV/(atom·ps), respectively.

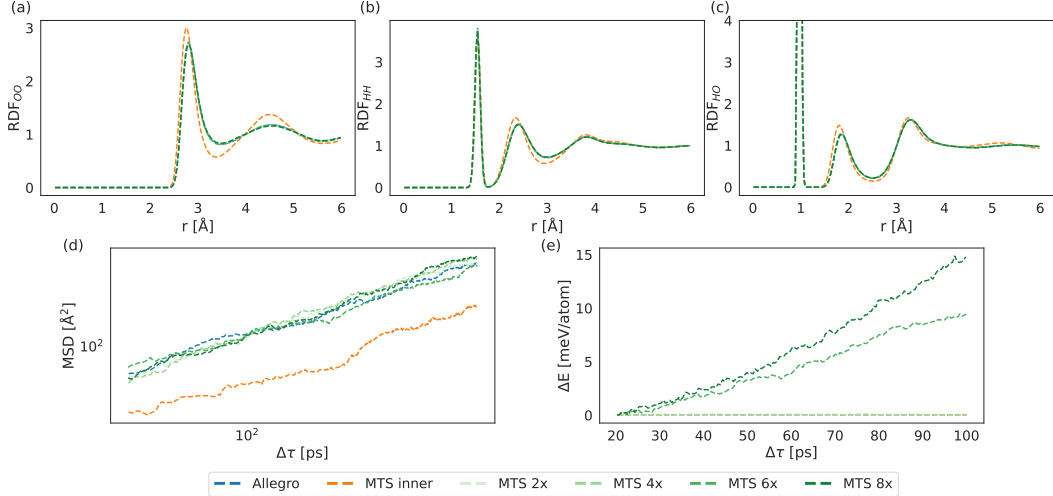


Figure 1: (a-c) O-O, H-H, and H-O RDFs of NVT simulations. (d) MSD of NVT simulations. (e) Total energy drift of NVE simulations. In the legend, MTS-Allegro is shortened for “MTS” along with the outer-inner time step ratio.

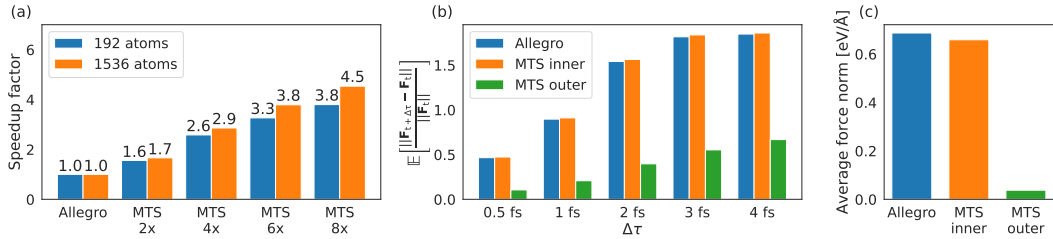


Figure 2: (a) Relative speedup factor compared to Allegro (1x in the plot). (b) Average relative frame-to-frame force difference in 0.5 fs time step Velocity Verlet MD for different models. (c) Average norm of forces for different models.

Speedup. Figure 2 (a) shows the relative speedup of MTS-Allegro compared to Allegro at two system sizes as measured in MD simulation². MTS-Allegro with a 4x outer time step, which preserves simulation-derived quantities and maintains energy conservation, achieves an MD speedup of 2.6x for a system size of 192 atoms and 2.9x for a system size of 1536 atoms. MTS-Allegro 6x and 8x, which obtain a further speedup, remain reliably stable in our experiments and faithfully reproduce RDFs and MSDs but do not maintain energy conservation. Such larger outer time step simulations may still find use in preliminary or other calculations whose requirements are less stringent.

Analysis of scale separation. We investigate how well MTS-Allegro separates scales by inspecting the time-smoothness and strength of the learned interactions. Figure 2 (b) shows the average relative frame-to-frame force difference (defined as $\mathbb{E}_t [\|\mathbf{F}_{t+\Delta\tau} - \mathbf{F}_t\| / \|\mathbf{F}_t\|]$ for each component) in 0.5 fs time step Velocity Verlet MD for different models. Compared to the conventional Allegro model and the inner model, the outer model of MTS-Allegro outputs forces with a much lower $\mathbb{E}_t [\|\mathbf{F}_{t+\Delta\tau} - \mathbf{F}_t\| / \|\mathbf{F}_t\|]$ for all $\Delta\tau$, indicating that it learns interactions that change much more slowly in time. Figure 2 (c) shows the average force norm (defined as $\mathbb{E}_t [\|\mathbf{F}_t\|]$) of different models. The outer model learns interactions that are much weaker in magnitude than Allegro and the inner model of MTS-Allegro. Weak and slow-varying interactions are exactly what allows for a longer time step. This analysis confirms the effectiveness of MTS-Allegro in learning scale separation.

²All simulations were run in LAMMPS [12] using pair_allegro on a single NVIDIA Tesla V100-PCIe 32GB GPU. The speed of Allegro is 0.51 and 0.072 ns/day for 192 and 1536 atoms, respectively.

5 Conclusion

We have developed a method to accelerate MLIP-driven MD simulations by learning a scale separation and using an MTS integrator. In an ab initio water system, our approach achieves around three times speedup without loss in accuracy on the potential energy or simulation-derived quantities. MTS integrators can also be used with more than two levels, and learning finer-grained scale separations with more than two MLIPs is a direction for future work. It is also possible to co-train different model architectures, such as kernel-based methods [18] and message-passing MLIPs [36], for accuracy-speed trade-offs. The presented technique promises a direction for significant practical speed gains when running MLIP-driven MD, including in large and complex systems.

Acknowledgments and Disclosure of Funding

We thank Simon Batzner, Cameron Owen, Yilun Xu, and Wujie Wang for valuable feedback and insightful discussions. X. F. and T. J. acknowledge support from the MIT-GIST collaboration. The work at Harvard was supported by Bosch Research, DOE Office of Basic Energy Sciences Award No. DE-SC0022199 and the Harvard University Materials Research Science and Engineering Center Grant No. DMR-2011754. A. M. was supported by DOE, Scientific Computing Research, Computational Science Graduate Fellowship under Award Number DE-SC0021110. Computing resources were provided by the MIT SuperCloud and Lincoln Laboratory Supercomputing Center and the Harvard University FAS Division of Science Research Computing Group.

References

- [1] William L Jorgensen, David S Maxwell, and Julian Tirado-Rives. Development and testing of the opls all-atom force field on conformational energetics and properties of organic liquids. *Journal of the American Chemical Society*, 118(45):11225–11236, 1996.
- [2] William L Jorgensen and Julian Tirado-Rives. Potential energy functions for atomic-level simulations of water and organic and biomolecular systems. *Proceedings of the National Academy of Sciences*, 102(19):6665–6670, 2005.
- [3] Helmut Grubmüller, Helmut Heller, Andreas Windemuth, and Klaus Schulten. Generalized verlet algorithm for efficient molecular dynamics simulations with long-range interactions. *Molecular Simulation*, 6(1-3):121–142, 1991.
- [4] Mark E Tuckerman and Bruce J Berne. Molecular dynamics in systems with multiple time scales: Systems with stiff and soft degrees of freedom and with short and long range forces. *The Journal of chemical physics*, 95(11):8362–8364, 1991.
- [5] MBBJM Tuckerman, Bruce J Berne, and Glenn J Martyna. Reversible multiple time scale molecular dynamics. *The Journal of chemical physics*, 97(3):1990–2001, 1992.
- [6] Bosco Garcia-Archilla, Jesús Maria Sanz-Serna, and Robert D Skeel. Long-time-step methods for oscillatory differential equations. *SIAM Journal on Scientific Computing*, 20(3):930–963, 1998.
- [7] Jesús A Izaguirre, Daniel P Catarello, Justin M Wozniak, and Robert D Skeel. Langevin stabilization of molecular dynamics. *The Journal of chemical physics*, 114(5):2090–2098, 2001.
- [8] Darryl D Humphreys, Richard A Friesner, and Bruce J Berne. A multiple-time-step molecular dynamics algorithm for macromolecules. *The Journal of Physical Chemistry*, 98(27):6885–6892, 1994.
- [9] Laxmikant Kalé, Robert Skeel, Milind Bhandarkar, Robert Brunner, Attila Gursoy, Neal Krawetz, James Phillips, Aritomo Shinozaki, Krishnan Varadarajan, and Klaus Schulten. NAMD2: greater scalability for parallel molecular dynamics. *Journal of Computational Physics*, 151(1):283–312, 1999.
- [10] Kevin J Bowers, Edmond Chow, Huafeng Xu, Ron O Dror, Michael P Eastwood, Brent A Gregersen, John L Klepeis, Istvan Kolossvary, Mark A Moraes, Federico D Sacerdoti, et al. Scalable algorithms for molecular dynamics simulations on commodity clusters. In *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*, pages 84–es, 2006.
- [11] David A Case, H Metin Aktulga, Kellon Belfon, Ido Ben-Shalom, Scott R Brozell, David S Cerutti, Thomas E Cheatham III, Vinícius Wilian D Cruzeiro, Tom A Darden, Robert E Duke, et al. *Amber 2021*. University of California, San Francisco, 2021.
- [12] 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. *Comp. Phys. Comm.*, 271:108171, 2022.
- [13] Nathan Luehr, Thomas E Markland, and Todd J Martínez. Multiple time step integrators in ab initio molecular dynamics. *The Journal of chemical physics*, 140(8), 2014.
- [14] Elisa Liberatore, Rocco Meli, and Ursula Rothlisberger. A versatile multiple time step scheme for efficient ab initio molecular dynamics simulations. *Journal of chemical theory and computation*, 14(6):2834–2842, 2018.
- [15] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.
- [16] Alireza Khorshidi and Andrew A Peterson. Amp: A modular approach to machine learning in atomistic simulations. *Computer Physics Communications*, 207:310–324, 2016.

- [17] Justin S Smith, Olexandr Isayev, and Adrian E Roitberg. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chemical science*, 8(4):3192–3203, 2017.
- [18] Stefan Chmiela, Alexandre Tkatchenko, Huziel E Sauceda, Igor Poltavsky, Kristof T Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science advances*, 3(5):e1603015, 2017.
- [19] Nongnuch Artrith, Alexander Urban, and Gerbrand Ceder. Efficient and accurate machine-learning interpolation of atomic energies in compositions with many species. *Physical Review B*, 96(1):014112, 2017.
- [20] Oliver T Unke and Markus Meuwly. A reactive, scalable, and transferable model for molecular energies from a neural network approach based on local information. *The Journal of chemical physics*, 148(24):241708, 2018.
- [21] Linfeng Zhang, Jiequn Han, Han Wang, Wissam Saidi, Roberto Car, et al. End-to-end symmetry preserving inter-atomic potential energy model for finite and extended systems. *Advances in Neural Information Processing Systems*, 31, 2018.
- [22] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and EJPRL Weinan. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Physical review letters*, 120(14):143001, 2018.
- [23] Roman Zubatyuk, Justin S Smith, Jerzy Leszczynski, and Olexandr Isayev. Accurate and transferable multitask prediction of chemical properties with an atoms-in-molecules neural network. *Science advances*, 5(8):eaav6490, 2019.
- [24] Dávid Péter Kovács, Cas van der Oord, Jiri Kucera, Alice EA Allen, Daniel J Cole, Christoph Ortner, and Gábor Csányi. Linear atomic cluster expansion force fields for organic molecules: beyond rmse. *Journal of chemical theory and computation*, 17(12):7696–7711, 2021.
- [25] Philipp Thölke and Gianni De Fabritiis. Equivariant transformers for neural network based molecular potentials. In *International Conference on Learning Representations*, 2021.
- [26] So Takamoto, Chikashi Shinagawa, Daisuke Motoki, Kosuke Nakago, Wenwen Li, Iori Kurata, Taku Watanabe, Yoshihiro Yayama, Hiroki Iriguchi, Yusuke Asano, et al. Towards universal neural network potential for material discovery applicable to arbitrary combination of 45 elements. *Nature Communications*, 13(1):1–11, 2022.
- [27] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.
- [28] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.
- [29] Justin S Smith, Roman Zubatyuk, Benjamin Nebgen, Nicholas Lubbers, Kipton Barros, Adrian E Roitberg, Olexandr Isayev, and Sergei Tretiak. The ani-1ccx and ani-1x data sets, coupled-cluster and density functional theory properties for molecules. *Scientific data*, 7(1):134, 2020.
- [30] Christian Devereux, Justin S Smith, Kate K Huddleston, Kipton Barros, Roman Zubatyuk, Olexandr Isayev, and Adrian E Roitberg. Extending the applicability of the ani deep learning molecular potential to sulfur and halogens. *Journal of Chemical Theory and Computation*, 16(7):4192–4202, 2020.
- [31] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2020.
- [32] Johannes Gasteiger, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. *Advances in Neural Information Processing Systems*, 34:6790–6802, 2021.

- [33] Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d molecular graphs. In *International Conference on Learning Representations*, 2021.
- [34] Oliver T Unke, Stefan Chmiela, Michael Gastegger, Kristof T Schütt, Huziel E Sauceda, and Klaus-Robert Müller. Spookynet: Learning force fields with electronic degrees of freedom and nonlocal effects. *Nature communications*, 12(1):1–14, 2021.
- [35] Cheol Woo Park, Mordechai Kornbluth, Jonathan Vandermause, Chris Wolverton, Boris Kozinsky, and Jonathan P Mailoa. Accurate and scalable graph neural network force field and molecular dynamics with direct force architecture. *npj Computational Materials*, 7(1):1–9, 2021.
- [36] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):1–11, 2022.
- [37] Albert Musaelian, Simon Batzner, Anders Johansson, Lixin Sun, Cameron J Owen, Mordechai Kornbluth, and Boris Kozinsky. Learning local equivariant representations for large-scale atomistic dynamics. *arXiv preprint arXiv:2204.05249*, 2022.
- [38] Oliver T Unke, Stefan Chmiela, Huziel E Sauceda, Michael Gastegger, Igor Poltavsky, Kristof T Schütt, Alexandre Tkatchenko, and Klaus-Robert Müller. Machine learning force fields. *Chemical Reviews*, 121(16):10142–10186, 2021.
- [39] Chi Chen and Shyue Ping Ong. A universal graph deep learning interatomic potential for the periodic table. *Nature Computational Science*, 2(11):718–728, 2022.
- [40] Bowen Deng, Peichen Zhong, KyuJung Jun, Janosh Riebesell, Kevin Han, Christopher J Bartel, and Gerbrand Ceder. Chgnet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nature Machine Intelligence*, pages 1–11, 2023.
- [41] Ruggero Lot, Franco Pellegrini, Yusuf Shaidu, and Emine Küçükbenli. Panna: Properties from artificial neural network architectures. *Computer Physics Communications*, 256:107402, 2020.
- [42] Xiang Fu, Zhenghao Wu, Wujie Wang, Tian Xie, Sinan Keten, Rafael Gomez-Bombarelli, and Tommi S. Jaakkola. Forces are not enough: Benchmark and critical evaluation for machine learning force fields with molecular simulations. *Transactions on Machine Learning Research*, 2023. Survey Certification.
- [43] Philipp Thölke and Gianni De Fabritiis. Torchmd-net: equivariant transformers for neural network based molecular potentials. *arXiv preprint arXiv:2202.02541*, 2022.
- [44] Samuel Schoenholz and Ekin Dogus Cubuk. Jax md: a framework for differentiable physics. *Advances in Neural Information Processing Systems*, 33:11428–11441, 2020.
- [45] Stefan Doerr, Maciej Majewski, Adrià Pérez, Andreas Kramer, Cecilia Clementi, Frank Noe, Toni Giorgino, and Gianni De Fabritiis. Torchmd: A deep learning framework for molecular simulations. *Journal of chemical theory and computation*, 17(4):2355–2363, 2021.
- [46] Ilyes Batatia, David P Kovacs, Gregor Simm, Christoph Ortner, and Gábor 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.
- [47] Hikaru Ibayashi, Taufeq Mohammed Razakh, Liqiu Yang, Thomas Linker, Marco Olguin, Shinnosuke Hattori, Ye Luo, Rajiv K Kalia, Aiichiro Nakano, Ken-ichi Nomura, et al. Allegrolegato: Scalable, fast, and robust neural-network quantum molecular dynamics via sharpness-aware minimization. In *International Conference on High Performance Computing*, pages 223–239. Springer, 2023.
- [48] Arthur Kosmala, Johannes Gasteiger, Nicholas Gao, and Stephan Günnemann. Ewald-based long-range message passing for molecular graphs. *arXiv preprint arXiv:2303.04791*, 2023.

- [49] Théo Jaffrelot Inizan, Thomas Plé, Olivier Adjoua, Pengyu Ren, Hatice Gökcan, Olexandr Isayev, Louis Lagardère, and Jean-Philip Piquemal. Scalable hybrid deep neural networks/polarizable potentials biomolecular simulations including long-range effects. *Chemical Science*, 14(20):5438–5452, 2023.
- [50] Mario Geiger and Tess Smidt. e3nn: Euclidean neural networks. *arXiv preprint arXiv:2207.09453*, 2022.
- [51] Albert Musaelian, Anders Johansson, Simon Batzner, and Boris Kozinsky. Scaling the leading accuracy of deep equivariant models to biomolecular simulations of realistic size, 2023.
- [52] Steve Plimpton. Particle-mesh ewald and rrespa for parallel molecular dynamics simulations. In *PPSC*. Citeseer, 1997.
- [53] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023.
- [54] Bingqing Cheng, Edgar A Engel, Jörg Behler, Christoph Dellago, and Michele Ceriotti. Ab initio thermodynamics of liquid and solid water. *Proceedings of the National Academy of Sciences*, 116(4):1110–1115, 2019.
- [55] Joost VandeVondele, Fawzi Mohamed, Matthias Krack, Jürg Hutter, Michiel Sprik, and Michele Parrinello. The influence of temperature and density functional models in ab initio molecular dynamics simulation of liquid water. *The Journal of chemical physics*, 122(1), 2005.
- [56] Weile Jia, Han Wang, Mohan Chen, Denghui Lu, Lin Lin, Roberto Car, E Weinan, and Linfeng Zhang. Pushing the limit of molecular dynamics with ab initio accuracy to 100 million atoms with machine learning. In *SC20: International conference for high performance computing, networking, storage and analysis*, pages 1–14. IEEE, 2020.
- [57] Mark Abraham, Andrey Alekseenko, Cathrine Bergh, Christian Blau, Eliane Briand, Mahesh Doijade, Stefan Fleischmann, Vytautas Gapsys, Gaurav Garg, Sergey Gorelov, Gilles Gouaillet, Alan Gray, M. Eric Irrgang, Farzaneh Jalalypour, Joe Jordan, Christoph Junghans, Prashanth Kanduri, Sebastian Keller, Carsten Kutzner, Justin A. Lemkul, Magnus Lundborg, Pascal Merz, Vedran Miletic, Dmitry Morozov, Szilárd Páll, Roland Schulz, Michael Shirts, Alexey Shvetsov, Bálint Soproni, David van der Spoel, Philip Turner, Carsten Uphoff, Alessandra Villa, Sebastian Wingbermühle, Artem Zhmurov, Paul Bauer, Berk Hess, and Erik Lindahl. Gromacs 2023.2 manual, July 2023.

A Supplementary Information

Table 2: Hyperparameters for Allegro and the outer model of MTS-Allegro.

Hyperparameter	Value
two body MLP latent dimensions	[128, 256, 512, 1024]
latent MLP latent dimensions	[1024, 1024, 1024]
number of interaction layers	2
radius cutoff	6 Å
maximum rotation order (l_{\max})	2
atom embedding multiplicity	32

Table 3: Hyperparameters for the inner model of MTS-Allegro.

Hyperparameter	Value
two body MLP latent dimensions	[64, 128, 256, 512]
latent MLP latent dimensions	[512, 512, 512]
number of interaction layers	1
radius cutoff	4 Å
maximum rotation order (l_{\max})	2
atom embedding multiplicity	32

Table 4: Hyperparameters for training. Both Allegro and Allegro-MTS use the same set of training hyperparameters (while M_{inner} is only applicable to Allegro-MTS).

Hyperparameter	Value
batch size	1
optimizer	Adam
initial learning rate	0.005
learning rate scheduler	ReduceLROnPlateau
learning rate patience	5 epochs
learning rate factor	0.5
early stopping learning rate	10^{-6}
λ_E	1
λ_F	2
inner only epochs (M_{inner} in Algorithm 2)	20 epochs