BoostMD: Accelerated Molecular Sampling Leveraging ML Force Field Features

Lars L. Schaaf[1,2] **Ilyes Batatia**[2] **Christoph Brunken**[1] **Thomas D. Barrett**[1] **Jules Tilly**[1]
[1]InstaDeep [2]University of Cambridge
{lls34, ib467}@cam.ac.uk {c.brunken, t.barrett, j.tilly}@instadeep.com

Abstract

Accurate modeling of atomic-scale processes, such as protein dynamics and catalysis, is a central challenge in computational structural biology, chemistry, and materials science. While machine learning force fields (MLFFs) have emerged as powerful tools, approaching quantum mechanical accuracy with promising generalisation capabilities, their application is hindered by prohibitive inference times, particularly for long timescale simulations of large systems required for many biological applications. In this work, we introduce BoostMD, a MLFF surrogate architecture, designed to mitigate this computational bottleneck. BoostMD leverages node features from previous molecular dynamics time steps to predict forces and energies, enabling the use of a smaller, faster model between evaluations of a large reference MLFF. The approach provides up to 8x speedup over the ground truth reference model. Testing on unseen dipeptides demonstrates that BoostMD accurately generalises and reproduces Boltzmann-distributed samples, making it a robust tool for efficient, long-timescale molecular simulations.

1 Introduction

Accurate modeling of atomic-scale processes, such as protein folding, catalysis, and carbon capture, is a long-standing challenge in computational biology, chemistry, and materials science [1–4]. Recent advancements in geometric deep learning have led to the development of machine learning force fields (MLFFs), which can predict atomic forces and energies based on 3D atomic configurations at near quantum mechanical accuracy [2, 5–11]. MLFFs promise cost-effective insight in the biomolecular structure, allowing *in-silico* screening of various desired properties. These models, trained on high-quality quantum mechanical data, typically scale linearly with system size, are more than 3 orders of magnitude faster for even small systems, while maintaining predictive accuracy [10, 11]. The emergence of generalizable MLFF foundation models, capable of modeling a wide range of molecules and materials, represents a significant breakthrough [1, 12]. Their transferable nature allows for direct application without a need to generate large expensive datasets or training from scratch. Furthermore, although trained on only small molecules the are sufficiently transferable for stable and accurate simulations of large peptides [1].

However, a main limitation of modern MLFFs is not their accuracy, but their inference time. Sampling algorithms such as Markov Chain Monte Carlo or Molecular Dynamics (MD) simulations need small time steps on the order of femtoseconds (10^{-15} s), to ensure reasonable acceptance rates and numerical stability, respectively. Sampling equilibrium configurations often requires hundreds of millions of such steps, to capture relevant conformational changes over longer timescales. With inference speeds on the order of 10^6 evaluations per day [1], MLFFs remain computationally prohibitive for many large-scale biological and material science applications.

In this work, we introduce BoostMD, a surrogate architecture designed to reduce the computational bottleneck of MLFFs. Typically, the majority of computational expense in MLFFs arises from generating expressive, atom-centered features of the 3D environment, while energy and forces are

38th Conference on Neural Information Processing Systems (NeurIPS 2024), D3S3 Workshop.

predicted using relatively simple and computationally inexpensive readout functions. BoostMD leverages node features computed at previous time steps to predict energies and forces for subsequent configurations. Hence, the full reference MLFF model is evaluated only every N steps, with a smaller, more efficient model operating in between. By leveraging previously computed features, a shallow and fast architectures remains accurate during these interim steps, significantly reducing the overall computational cost while maintaining accuracy. Our contributions are as follows:

- We introduce, BoostMD, **a new approach for accelerating** ML force fields, by using previously computed node features to predict the energy and forces on atoms of subsequent simulation steps.
- We describe the **design space** of possible BoostMD models, both for training and sampling. We empirically test and contrast various fundamental architectural choices.
- We present the first BoostMD model, which provides more than eight times speedup. We show that it is **transferable** to unseen molecules and capable of accurately sampling equilibrium structures with the right probability distribution.

2 Background and Notation

Equivariant message passing neural networks Recent progress in equivariant message passing neural networks [10, 11], have allowed for unprecedentedly accurate MLFFs. In these models, a graph is constructed from a point cloud of atoms by connecting atoms within a cutoff distance. For equivariant models, the messages passed between nodes (atoms) encode the atoms geometric environment and transform with rotations in a controlled way. After a set number of message passing steps, the node features pass through a readout that predicts the atom centered energy (a scalar quantity). The majority of the computational cost originates from creating many-body, descriptive node features. The final readout are small MLPs or linear layers [2, 10, 11].

Irreducible representations and tensor products When learning equivariant features from atomic positions, it is helpful to use the irreducible representations of the $\mathrm{SO}(3)$ group. The spherical harmonics Y_m^l, where l is the degree and m is the order, form a basis for the vector spaces on which the irreducible representations act. Nodes can thus have scalar features ($l = 0$) that are invariant under rotation or higher-order features ($l > 0$), such as vectors, that transform equivariantly with rotations.

In non-geometric machine learning we can use tensor products of two features $(A \otimes B)_{ijkl} = A_{ij}B_{jk}$ to generate higher dimensional tensors. For E(3) equivariant features, we need to map the output back to the irreducible representation (irreps). Furthermore, the coefficients for the weights are restricted as to maintain the underlying equivariance. Spherical tensors, like the spherical harmonics are indexed by the degree l and order m. Assuming two spherical tensors $A_{l_1}^{m_1}$ and $B_{l_2}^{m_2}$ a learnable fully connected tensor product in irreducible representations is defined as

$$D_{l_3}^{m_3} = (\boldsymbol{A} \overset{\boldsymbol{\alpha}}{\otimes} \boldsymbol{B})_{l_3}^{m_3} := \sum_{l_1 m_1, l_2 m_2} C_{l_1 m_1, l_2 m_2}^{l_3 m_3} F_{l_1 l_2 l_3}(\boldsymbol{\alpha})\, A_{l_1}^{m_1} B_{l_2}^{m_2}, \tag{1}$$

where $F_{l_1 l_2 l_3}$ is a multi-layer perceptron (MLP) with scalar inputs $\boldsymbol{\alpha}$ and $C_{l_1, m_1; l_2, m_2}^{l_3 m_3}$ are the Clebsch–Gordan coefficients. This is, for example, how the position is combined with the sending node features in equivariant MPNNs. For more details please see the *e3nn* paper [13] or classic literature [14].

3 BoostMD

The BoostMD model predicts the node features, h_i, and energy, E_i, of atom i at position r_i, given a reference node feature H_i^{ref} for a configuration with positions r_i^{ref}. This task can be approached with many different architectures. We build on the equivariant MACE [11] blocks, which provide an efficient way to obtain many-body equivariant features. We use the tensor product notation ($\otimes$) introduced in Section 2 without explicitly writing the indices. Please see Appendix B for a more explicit equations and dimensions of all tensors.

Set Reference Frame To be transitionally and rotational equivariant, the reference frame is transformed as visualised in Figure 2. Firstly, the origin is shifted to the central atom current position r_i. Then, the reference is rotated by $\bar{\theta}$, as to minimise the L2 norm of displacements between the neighbouring atoms' reference and current position,

method	model size	L^{ref}	mode	speedup	E (meV)	F (meV/Å)
BoostMD Model	XS	0	feats	8.4	1.30	64.5
	XS	0	energy	**8.6**	0.84	75.3
	XS	1	feats	2.3	0.64	57.6
	XS	1	energy	2.3	**0.63**	**56.9**
MACE	XS^+	-	-	5.6	1.80	121.1

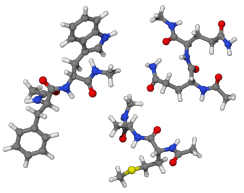

Table 1: **Comparing design choices on dipeptide dataset.** BoostMD models trained on a dipeptide MD dataset subselected from SPICE [16]. The energy per atom and force RMSE are computed with respect to the MACE-OFF23-medium ground truth. The reference model, MACE-OFF23-M, achieves an error of 0.85 meV/atom compared to DFT, higher than many of the boost models. Speedups are compared to the ground truth model. For details on hyperparameters corresponding to size (XS, XS^+) and the timing see Appendix D.1 and D.2 respectively.

$$r_{ji} = r_j - r_i, \qquad r^{\text{ref}}_{ji} = r^{\text{ref}}_j - r^{\text{ref}}_i, \qquad x_{ji} = r_{ji} - R_{\bar{\theta}_i}(r^{\text{ref}}_{ji}), \tag{2}$$

where x_{ij} encodes the change in position and r_{ij} encodes the current displacement. The optimal rotation $\bar{\theta}_i$ is efficiently computed using the Kabsch algorithm [15] and ensures rotational equivariance. Further detail and its relevance for conservation of momentum can be found in Appendix B.1.

Equivariant message passing We now construct a complete basis for functions of the change in positions x_{ij}, current positions r_{ij} and the sender's reference node features H^{ref}_i. The initial edge features $\mathfrak{h}_{ji}$ are then summed across neighbours j, to create the BoostMD A-Basis.

$$\mathfrak{h}_{ji} = Y^{m_1}_{l_1}(\hat{\boldsymbol{r}}_{ji}) \overset{|\mathbf{x}||\mathbf{r}|}{\otimes} Y^{m_2}_{l_2}(\hat{\boldsymbol{x}}_{ji}) \otimes R_{\bar{\theta}_i}(\mathbf{H}^{\text{ref}}_j) \qquad \mathbf{A}_i = \left(\textstyle\sum_{j \in N(i)} \mathfrak{h}^{(t)}_{ij}\right) \tag{3}$$

We subsequently use the MACE [11] product basis to construct many body messages, $\mathbf{m}$, from the two body A-basis. Finally we use a readout MLP to map internal BoostMD features to either (1) the energies E_i directly or (2) the node features of the true MLFF at position r_i. For BoostMD with readout-mode (2), the original reference's readout is used to predict the final atom centered energy.

3.1 BoostMD at inference

In BoostMD, the full reference MLFF model is evaluated every N steps, while a smaller, more efficient model operates in between. This new architecture can be used to accelerate molecular dynamics in various ways, each with different impacts on performance:

Serial evaluation of reference and BoostMD model On a single GPU, the MD simulation pauses while the reference model is evaluated. After the reference features are calculated, the BoostMD model takes N steps, then waits for the next reference calculation. This approach trades some of the speedup for the benefit of directly incorporating the reference model's true forces into the MD run.

Parallel evaluation of reference and BoostMD model Alternatively, the reference calculations could be performed on another GPU in parallel. The BoostMD model does not have to wait for the reference model evaluation, resulting in greater acceleration.
There are many more ways BoostMD models can be used at inference such as having adaptive step sizes based on observed errors or changes in receptive field. This broad design space underscores the potential of this approach for various applications.

Symmetries and Conservation Laws The resulting trajectory from a BoostMD run does not conserve energy exactly. This is because the energy at a given position is dependant on the reference configuration. Consequently it is possible to have a different energy predicted for the same configuration depending on the reference. However, in-between time-steps of the reference MLFF the method is energy, momentum and angular momentum conserving due to the reference framing. There are multiple methods, such as QM/MM calculations with adaptive QM regions that are not exactly energy conserving [17]. It has been shown that the thermostat may ensure ergodic sampling at the correct temperature even in the presence of energy conservation violation [18].

4 Experiments

In this section we train BoostMD models to accelerate a foundation model for organic molecules, namely MACE-OFF23 [1]. The force field has been shown to replicate folding dynamics of peptides

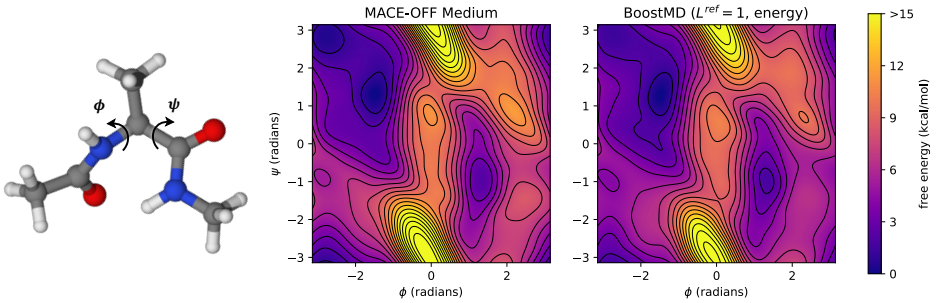

Figure 1: **Free energy surface of unseen alanine-dipeptide** Comparison of the samples obtained by running ground truth MD and boostMD. The free energy of the Ramachandran plot, is directly related to the marginalized Boltzmann distribution $\exp\left[-F(\phi,\psi)/k_BT\right]$. The reference model is evaluated every 10 steps. Both simulations are run for 5 ns (5×10^6 steps).

and accurately predict experimental observable relevant for drug discovery. We show how choices in the BoostMD architecture affect accuracy and performance, illustrating up to 8x speedup and stable dynamics.

Dipeptide dataset: the building blocks of proteins To assess the performance gains of various BoostMD models, we trained them on a dataset of dipeptides, generated by running molecular dynamics (MD) simulations using the MACE-OFF23 medium foundation model [1]. The training set dynamics are initialised from the dipeptide subset of the the SPICE dataset [16]. We observe that single-layer BoostMD models provide speedups exceeding $8\times$, as shown in Table 1 with low errors. With a single layer, the BoostMD model has a receptive field of 5Å, in principle allowing for more efficient parallelisation across GPUs using domain decomposition.

MACE-OFF23-M achieves an error of 0.85 meV/atom compared to DFT, with BoostMD models showing similar or lower accuracy relative to MACE-OFF23-M. We explored models, using only invariant reference node features ($L^{\text{ref}} = 0$) as well as those incorporating equivariant features ($L^{\text{ref}} = 1$). The inclusion of equivariant features improves accuracy at some computational cost. Additionally, we examined the two readout strategies from Section 3, finding that directly predicting energy changes is both marginally more accurate and faster than predicting changes in node features.

A natural alternative to training a BoostMD model would be to simply train a faster model from scratch. Motivated by this, we trained a single layer MACE model (XS^+), chosen to provide the same order of speed-up over the reference MACE-OFF23-M model as achieved by BoostMD. However, we find that the energy and force's are less accurate than those achieved by BoostMD, highlighting the advantages of biasing on previous features.

Sampling of unseen Dipeptide Beyond low RMSE values, the true test of BoostMD is its ability to accurately sample equilibrium configurations at a given temperature. We found that BoostMD is stable during MD simulations, remaining robust even after $10\,\text{ns}$ (10^7 steps). When tested on an unseen molecule, alanine dipeptide, BoostMD accurately sampled molecular configurations. Figure 1, shows the free energy, F, as a function of the backbone angle. The typical Ramachandran plot, is directly related to the marginalized Boltzmann distribution $\exp\left[-F(\phi,\psi)/k_{\text{B}}T\right]$, where T is the temperature. This demonstrates that BoostMD can efficiently and accurately sample configurations, making it a valuable tool for long-timescale molecular simulations.

5 Discussion

We introduce BoostMD, a surrogate architecture to accelerate molecular sampling with machine learning force fields (MLFFs). BoostMD leverages node features from a reference MLFF model at previous steps, allowing a smaller, efficient model to predict energies and forces in subsequent steps. This significantly reduces computational costs, achieving up to a $8\times$ speedup with minimal accuracy loss. Notably, BoostMD retains its performance on unseen systems, accurately sampling the Boltzmann distribution of unseen dipeptides. Future work will involve implementing adaptive inference schemes and scaling training to larger datasets, further enhancing BoostMDs ability to accelerate MLFF molecular simulations. We hope that BoostMD will enabling routine, accurate, large scale simulations to accelerate computational screening of biological compounds for catalytic and medical applications.

References

1. Kovács, D. P. *et al. MACE-OFF23: Transferable Machine Learning Force Fields for Organic Molecules* Dec. 29, 2023. arXiv: 2312.15211[physics]. http://arxiv.org/abs/2312.15211 (2024).
2. Behler, J. & Parrinello, M. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Phys. Rev. Lett.* **98,** 146401. ISSN: 0031-9007, 1079-7114. https://link.aps.org/doi/10.1103/PhysRevLett.98.146401 (2024) (Apr. 2, 2007).
3. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *nature* **596,** 583–589 (2021).
4. Chen, B. W., Xu, L. & Mavrikakis, M. Computational methods in heterogeneous catalysis. *Chemical Reviews* **121,** 1007–1048 (2020).
5. Bartók, A. P., Payne, M. C., Kondor, R. & Csányi, G. Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons. *Phys. Rev. Lett.* **104,** 136403. ISSN: 0031-9007, 1079-7114. https://link.aps.org/doi/10.1103/PhysRevLett.104.136403 (2024) (Apr. 1, 2010).
6. Schütt, K. *et al. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions* in *Advances in Neural Information Processing Systems* **30** (Curran Associates, Inc., 2017). https://proceedings.neurips.cc/paper/2017/hash/303ed4c69846ab36c2904d3ba8573050-Abstract.html (2024).
7. Shapeev, A. V. Moment Tensor Potentials: A Class of Systematically Improvable Interatomic Potentials. *Multiscale Model. Simul.* **14,** 1153–1173. ISSN: 1540-3459, 1540-3467. http://epubs.siam.org/doi/10.1137/15M1054183 (2024) (Jan. 2016).
8. Bochkarev, A. *et al.* Efficient parametrization of the atomic cluster expansion. *Phys. Rev. Materials* **6,** 013804. ISSN: 2475-9953. https://link.aps.org/doi/10.1103/PhysRevMaterials.6.013804 (2024) (Jan. 24, 2022).
9. Nigam, J., Pozdnyakov, S. & Ceriotti, M. Recursive evaluation and iterative contraction of N-body equivariant features. *The Journal of chemical physics* **153.** Publisher: AIP Publishing. https://pubs.aip.org/aip/jcp/article/153/12/121101/1023428 (2024) (2020).
10. Batzner, S. *et al.* E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat Commun* **13,** 2453. ISSN: 2041-1723. https://www.nature.com/articles/s41467-022-29939-5 (2024) (May 4, 2022).
11. Batatia, I., Kovács, D. P., Simm, G. N. C., Ortner, C. & Csányi, G. *MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields* Jan. 26, 2023. arXiv: 2206.07697[cond-mat,physics:physics,stat]. http://arxiv.org/abs/2206.07697 (2024).
12. Batatia, I. *et al. A foundation model for atomistic materials chemistry* Mar. 1, 2024. arXiv: 2401.00096[cond-mat,physics:physics]. http://arxiv.org/abs/2401.00096 (2024).
13. Geiger, M. & Smidt, T. *e3nn: Euclidean Neural Networks* July 18, 2022. arXiv: 2207.09453[cs]. http://arxiv.org/abs/2207.09453 (2024).
14. Bourbaki, N. *Lie groups and Lie algebras* 247-267 (Elements of the History of Mathematics, 1994).
15. Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Cryst A* **32.** Publisher: International Union of Crystallography, 922–923. ISSN: 0567-7394. https://journals.iucr.org/a/issues/1976/05/00/a12999/ (2024) (Sept. 1, 1976).
16. Eastman, P. *et al.* SPICE, A Dataset of Drug-like Molecules and Peptides for Training Machine Learning Potentials. *Sci Data* **10.** Publisher: Nature Publishing Group, 11. ISSN: 2052-4463. https://www.nature.com/articles/s41597-022-01882-6 (2024) (Jan. 4, 2023).
17. Várnai, C., Bernstein, N., Mones, L. & Csányi, G. Tests of an Adaptive QM/MM Calculation on Free Energy Profiles of Chemical Reactions in Solution. *J. Phys. Chem. B* **117,** 12202–12211. ISSN: 1520-6106, 1520-5207. https://pubs.acs.org/doi/10.1021/jp405974b (2024) (Oct. 10, 2013).
18. Jones, A. & Leimkuhler, B. Adaptive stochastic methods for sampling driven molecular systems. *The Journal of Chemical Physics* **135,** 084125. ISSN: 0021-9606, 1089-7690. https://pubs.aip.org/jcp/article/135/8/084125/72033/Adaptive-stochastic-methods-for-sampling-driven (2024) (Aug. 28, 2011).

19. Bussi, G. & Laio, A. Using metadynamics to explore complex free-energy landscapes. *Nat Rev Phys* **2,** 200–212. ISSN: 2522-5820. https://www.nature.com/articles/s42254-020-0153-0 (2024) (Mar. 6, 2020).
20. Kästner, J. Umbrella sampling. *WIREs Computational Molecular Science* **1.** _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.66, 932–942. ISSN: 1759-0884. https://onlinelibrary.wiley.com/doi/abs/10.1002/wcms.66 (2024) (2011).
21. Berne, B. J. *Molecular Dynamics in Systems with Multiple Time Scales: Reference System Propagator Algorithms* in *Computational Molecular Dynamics: Challenges, Methods, Ideas* (eds Deuflhard, P. *et al.*) (Springer, Berlin, Heidelberg, 1999), 297–317. ISBN: 978-3-642-58360-5.
22. Tuckerman, M., Berne, B. J. & Martyna, G. J. Reversible multiple time scale molecular dynamics. *The Journal of Chemical Physics* **97,** 1990–2001. ISSN: 0021-9606. https://doi.org/10.1063/1.463137 (2024) (Aug. 1, 1992).
23. Fu, X., Musaelian, A., Johansson, A., Jaakkola, T. & Kozinsky, B. Learning Interatomic Potentials at Multiple Scales. http://arxiv.org/abs/2310.13756 (Oct. 2023).
24. Tabak, E. G. & Turner, C. V. A Family of Nonparametric Density Estimation Algorithms. *Communications on Pure and Applied Mathematics* **66.** _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.21423, 145–164. ISSN: 1097-0312. https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.21423 (2024) (2013).
25. Noé, F., Olsson, S., Köhler, J. & Wu, H. *Boltzmann Generators – Sampling Equilibrium States of Many-Body Systems with Deep Learning* July 12, 2019. arXiv: 1812.01729[cond-mat,physics:physics,stat]. http://arxiv.org/abs/1812.01729 (2024).
26. Klein, L. & Noé, F. *Transferable Boltzmann Generators* June 20, 2024. arXiv: 2406.14426[physics,stat]. http://arxiv.org/abs/2406.14426 (2024).
27. Midgley, L. I., Stimper, V., Simm, G. N. C., Schölkopf, B. & Hernández-Lobato, J. M. *Flow Annealed Importance Sampling Bootstrap* Mar. 7, 2023. arXiv: 2208.01893[cs,q-bio,stat]. http://arxiv.org/abs/2208.01893 (2024).
28. Midgley, L. I. *et al. SE(3) Equivariant Augmented Coupling Flows* Mar. 5, 2024. arXiv: 2308.10364[physics]. http://arxiv.org/abs/2308.10364 (2024).
29. Coretti, A., Falkner, S., Weinreich, J., Dellago, C. & von Lilienfeld, O. A. *Boltzmann Generators and the New Frontier of Computational Sampling in Many-Body Systems* 2024. arXiv: 2404.16566[physics]. http://arxiv.org/abs/2404.16566 (2024).
30. Rotskoff, G. M. Sampling thermodynamic ensembles of molecular systems with generative neural networks: Will integrating physics-based models close the generalization gap? *Current Opinion in Solid State and Materials Science* **30,** 101158. ISSN: 1359-0286. https://www.sciencedirect.com/science/article/pii/S135902862400024X (2024) (June 1, 2024).
31. Elijošius, R. *et al.* Zero Shot Molecular Generation via Similarity Kernels. http://arxiv.org/abs/2402.08708 (Feb. 2024).
32. Klein, L. *et al. Timewarp: Transferable Acceleration of Molecular Dynamics by Learning Time-Coarsened Dynamics* Dec. 1, 2023. arXiv: 2302.01170[cond-mat,physics:physics,stat]. http://arxiv.org/abs/2302.01170 (2024).
33. Morrow, J. D. & Deringer, V. L. Indirect Learning of Interatomic Potentials for Accelerated Materials Simulations. *The Journal of Chemical Physics* **157,** 104105. ISSN: 0021-9606, 1089-7690. arXiv: 2111.11120[cond-mat,physics:physics]. http://arxiv.org/abs/2111.11120 (2024) (Sept. 14, 2022).
34. Maday, Y. & Turinici, G. A parareal in time procedure for the control of partial differential equations. *Comptes Rendus Mathematique* **335,** 387–392. ISSN: 1631-073X. https://www.sciencedirect.com/science/article/pii/S1631073X02024676 (2024) (Jan. 1, 2002).
35. Gorynina, O., Legoll, F., Lelievre, T. & Perez, D. *Combining machine-learned and empirical force fields with the parareal algorithm: application to the diffusion of atomistic defects* Dec. 20, 2022. arXiv: 2212.10508[cs,math]. http://arxiv.org/abs/2212.10508 (2024).
36. Darby, J. P. *et al.* Tensor-Reduced Atomic Density Representations. *Phys. Rev. Lett.* **131,** 028001. ISSN: 0031-9007, 1079-7114. https://link.aps.org/doi/10.1103/PhysRevLett.131.028001 (2024) (July 13, 2023).

Appendices

A Related Work

Enhanced sampling techniques Enhanced sampling methods, such as metadynamics [19] and umbrella sampling [20], introduce controlled biases to overcome energy barriers and access rare events that are difficult to observe with standard molecular dynamics (MD). Samples can be reweighted to recover unbiased equilibrium observables. BoostMD can be combined with these methods, compounding the efficiency gains and further increasing its practical relevance.

Reference system propagator algorithms In traditional non-ML force fields it is possible to evaluate a part of the force field every time step, while other, more expensive parts of the force field are only evaluated every x time steps [21, 22]. The core idea is that some terms (eg. shortrange bonded interactions) change more rapidly than others (long-range electorstatics). Recently, a similar approach has been applied to ML force fields [23]. This approach is different from BoostMD, as it does not bias the potential on the node features of previous time-steps. BoostMD's speedup is independent of the interaction range and can be combined with such range separated efforts.

Generative modeling Generative models, including diffusion models and normalizing flows [24], offer an alternative to sequential approaches by directly generating samples from a target distribution [25–31]. Although promising, these models often lack transferability, require extensive data generation for each specific problem, or have only demonstrated effectiveness on small systems with lower dimensionality [25, 26]. Consequently, they have yet to surpass the performance of MLFFs combined with traditional sampling methods. Unlike MLFFs, like BoostMD, it is not trivial to combine these approaches with enhanced sampling techniques [32].

Others It is possible to train a smaller model on synthetic data generated by a larger teacher model, to obtain computational speedup [33]. Additionally, there exist parallel-in-time integrator methods, where computational cost for a sequential MD trajectory can be parallelised, such as the parareal approach [34, 35].

B Detailed equations and BoostMD architecture

In this section we provide the details of the BoostMD architecture. The BoostMD model predicts the node features, $\mathbf{h}_i$, and energy, E_i, of atom i at position $\mathbf{r}_i$, given a set of reference node feature $\mathbf{H}_j^{\text{ref}}$

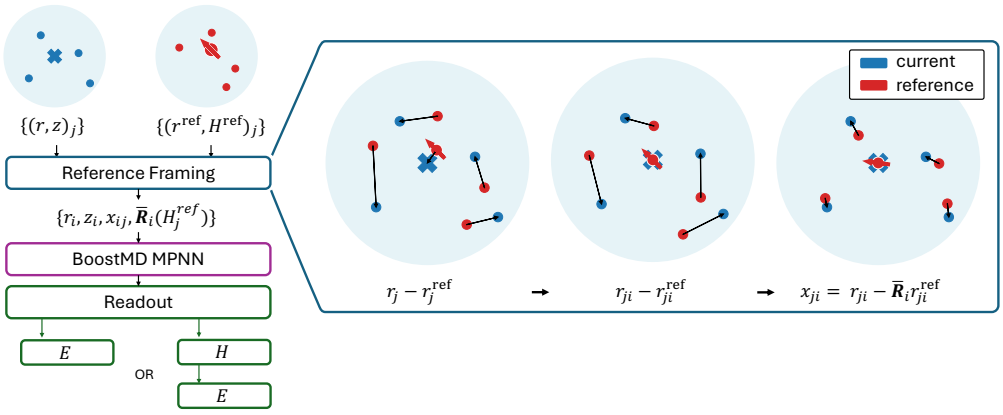

Figure 2: **BoostMD and reference framing** Showing the steps of BoostMD models, highlighting the reference framing step. The reference node features and reference positions are transformed to make BoostMD translationally and rotationally equivariant between steps as detailed in equation 2. The figures show the receptive field of an atom i with neighbours j, showing both the current (blue) and reference (red) positions. The central red arrow represents an equivariant reference feature of atom i, while the black arrows show the vectors associated with the label underneath each image.

for a configuration with positions $\mathbf{r}_j^{\text{ref}}$, where j is within the neighbourhood $\mathcal{N}(i)$ of atom i. This task can be approached with many different architectures. The first BoostMD model we present in this paper makes use of blocks from the `e3nn` [13] and `MACE`[11] library.

We start by motivating the choice of coordinates, ie reference framing. We then provide equations for the entire architecture.

B.1 Reference framing

To ensure translational and rotational equivariance, the reference frame is transformed, as visualised in Figure 2. We refer to the sending and receiving nodes/atoms as j and i are the respectively.

Firstly, we determine the displacement between atoms j and i for both the current position, $\mathbf{r}_{ij}$ and the reference position, $\mathbf{r}_{ji}^{\text{ref}}$,

$$\mathbf{r}_{ji} = \mathbf{r}_j - \mathbf{r}_i, \qquad \mathbf{r}_{ji}^{\text{ref}} = \mathbf{r}_j^{\text{ref}} - \mathbf{r}_i^{\text{ref}} \tag{4}$$

This ensures translational equivalence, which guarantees conservation of momentum in between BoostMD steps and that the forces sum to zero.

To handle environment changes that correspond purely to rotations, parameterizing changes in displacement as $\mathbf{r}_{ji} - \mathbf{r}_{ji}^{\text{ref}}$ is insufficient because rotational changes should directly map to rotated node features. To ensure exact equivariance, we compute the optimal rotation matrix $\bar{\mathbf{R}}_i$ that best aligns the reference configuration with the current configuration:

$$\bar{\mathbf{R}}_i = \arg\min_{\mathbf{R}_i} \sum_{j \in \mathcal{N}(i)} \left\| \mathbf{r}_{ji} - \mathbf{R}_i \mathbf{r}_{ji}^{\text{ref}} \right\|^2 .$$

The optimal rotation $\bar{\mathbf{R}}_i$ is efficiently computed using the Kabsch algorithm [15]. The rotation is also applied to the reference node features and guarantees rotational equivariance between BoostMD steps. The change in positions, $\mathbf{x}_{ij}$, is hence defined as

$$\mathbf{x}_{ji} = \mathbf{r}_{ji} - \bar{\mathbf{R}}_i \mathbf{r}_{ji}^{\text{ref}}, \tag{5}$$

B.2 First layer-features

Firstly we take a tensor product between the changes in displacement with learnable weights, defining the $\mathbf{X}$ basis:

$$X_{ji}^{l_3 m_3} = \sum_{l_1 m_1, l_2 m_2} C_{l_1 m_1, l_2 m_2}^{l_3 m_3} F_{l_1 l_2 l_3}\left(r_{ji}, x_{ji}\right) Y_{l_1}^{m_1}\left(\hat{\boldsymbol{r}}_{ji}\right) Y_{l_2}^{m_2}\left(\hat{\boldsymbol{x}}_{ji}\right), \tag{6}$$

where $F_{l_1 l_2 l_3}$ is a multi-layer perceptron (MLP) of the inter-atomic distance $\mathbf{r}_{ji}$ and the size of the change in displacement $\mathbf{x}_{ji}$. Before being passed to the MLP, the distances are embedded using Gaussian basis functions. We now perform a tensor-reduced tensor product [36] with the reference features,

$$\mathbf{A}_{i,kl_3m_3}^{(1)} = \sum_{l_1 m_1, l_2 m_2} C_{l_1 m_1, l_2 m_2}^{l_3 m_3} \sum_{j \in \mathcal{N}(i)} F'_{kl_1 l_2 l_3}\left(r_{ji}, x_{ji}, \mathbf{H}_{j,l=0}^{\text{ref}}\right) X_{ji}^{l_3 m_3} \sum_{\tilde{k},t} W_{k\tilde{k}l_2}^{(t)} D_{\bar{\mathbf{R}}i} H_{j,\tilde{k}l_2 m_2}^{\text{ref},(t)} \tag{7}$$

where $D_{\bar{\mathbf{R}}i}$ is the Wigner-D matrix corresponding to the previously determined optimal rotation, $\mathbf{H}_{j,\tilde{k}l_2m_2}^{\text{ref},(t)}$ are the reference node features of the tth layer of the reference model and F' is defined similar to F. Summing over all neighbours $\mathcal{N}(i)$, we can now employ the density trick and take tensor powers of the $\mathbf{A}$-basis to create many-body features. Here we directly use the MACE [11] architecture (equ. 10) to construct the $\mathbf{B}$-basis,

$$B_{i,\eta_\nu kLM}^{(1)} = \sum_{\boldsymbol{lm}} \mathcal{C}_{\eta_\nu,\mathbf{lm}}^{LM} \prod_{\xi=1}^{\nu} \sum_{\tilde{k}} w_{k\tilde{k}l_\xi}^{(1)} A_{i,\tilde{k}l_\xi m_\xi}^{(1)}, \quad \boldsymbol{lm} = \left(l_1 m_1, \ldots, l_\nu m_\nu\right), \tag{8}$$

where ν is the correlation order and $\mathcal{C}^{LM}_{\eta_\nu, l\boldsymbol{m}}$ are the generalised Clebsch-Gordan coefficients. For more details please see the MACE [11] paper directly. The $\mathbf{B}$-basis is used to update the node features,

$$m^{(1)}_{i,kLM} = \sum_{\nu} \sum_{\eta_\nu} W^{(1)}_{z_i kL,\eta_\nu} B^{(1)}_{i,\eta_\nu kLM}, \qquad h^{(2)}_{i,kLM} = \sum_{\tilde{k}} W^{(1)}_{kL,\tilde{k}} m^{(t)}_{i,\tilde{k}LM}, \tag{9}$$

where $\mathbf{h}$ are the internal BoostMD features. The architecture for the next layers of the BoostMD model are identical to that of MACE [11]. While it is possible to run multi-layer BoostMD models, for speed purposes this is rarely of interest. Consequently, all presented results are performed with single layer models.

B.3 Readout

Finally, BoostMD can either be used to (1) directly predict changes in energies E_i or (2) changes in the node features of the true MLFF $\mathbf{h}_i$ at position $\mathbf{r}_i$. Assuming the BoostMD model has t' layers, the final energy depends on the readout mode:

1. **Direct energy readout** The reference model has an energy associated with each of its layers t. We perform a readout on each of the BoostMD layers t' for each reference layer energy t,

$$E^{(t)}_i = E^{\text{ref},(t)}_i + \sum_{t'} \texttt{MLP}^{(t')}(\mathbf{h}^{(t')}_i) \sum_j |x_{ij}|, \tag{10}$$

   where the multiplicative factor $|x_{ij}|$, ensures that when there is no change in environment, then there is no change in predicted energy.

2. **Reference node feature readout** Here we predict the changes to the reference's models node features.

$$\mathbf{h}^{(t)}_i = R_{\bar{\theta}_i}(\mathbf{H}_{i,k}) + \sum_{t'} \texttt{MLP}^{(t')}(\mathbf{h}^{(t')}_i) \sum_j |x_{ij}| \tag{11}$$

   The original readout of the reference model $\mathcal{R}_t$ can then be used to compute the energy as

$$E^{(t)}_i = \mathcal{R}_t \left(\mathbf{h}^{(t)}_i \right). \tag{12}$$

The final total energy is simply a sum over layer energies and atoms, $\sum_{t,i} E^{(t)}_i$. Forces can be determined using backpropagation. As previously mentioned, for speed reasons all experiments are performed with a single layer BoostMD model.

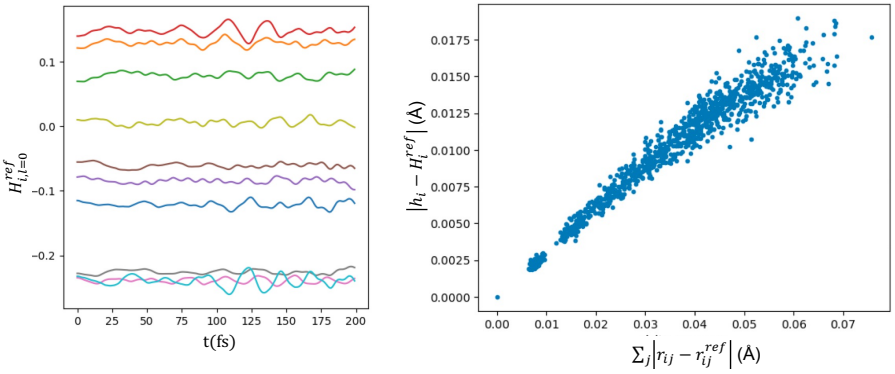

Figure 3: **Node feature properties** Showing the fluctuations of the reference node features as a function of simulation time (**a**) and the change in node features as a function of change in position inside the atomic environment.

C Node feature and their properties

The equivariant node features of large foundation models are highly expressive low-dimensional representations of the many-body atomic environment. Indeed, similarity kernels of reference descriptors have already been used for zero-shot molecular generation [31]. Figure 3a, shows how the node features change as a function of time during a typical MD simulation. It is clear that they vary very smoothly and the changes remain small with periodic fluctuations. This motivates to goal of predicting node features with the BoostMD architecture. Furthermore the magnitude of the change in node features is proportional to the change in the atomic environment as visible in Figure 3, motivating the multiplicative factor in readout equations B.3 and 12.

D Details on Experiments

D.1 Model hyper-parameters

The aim is to produce a fast architecture while maintaining sufficient accuracy. We restrict ourselves to a single layer BoostMD model as this not only reduces the model cost, but would also allow for more parallelisation. For big simulations, local models can be partitioned into smaller overlapping regions and evaluated in parallel on multiple GPUs. The size of the overlap needed depends on the total receptive field. A single layer reduces the receptive field from 10Å to 5Å significantly increasing its prallelisation abilities.

For the BoostMD models, we define the XS size as a single model, with correlation order 2, compared to 3 for the reference foundation model. Furthermore, the maximum order of the spherical harmonics is set to 2, compared to 3 for the MACE-OFF-M model. We use 125 channels for the BoostMD model. As the BoostMD architecture requires one additional edge tensor product in equation 6, comapred to the MACE architecture, a MACE model is faster than a BoostMD model with the same hyperparameters. For a fair comparison, we hence give the compared MACE model additional flexibility by increasing the number of channels to 256 (XS^+). As visible in Table 1, even with this added flexibility the MACE model underperforms a BoostMD model, which depends on previously computed node features. All experimental results do not use the rotational reference framing during training or inference, due to the associated computational cost.

D.2 Timings

Timing measurements were conducted on a system consisting of a 3BPA molecular box with a total of 3,375 atoms. This large system size is representative of typical biomolecular simulations and minimizes the impact of PyTorch overhead on the measured runtime. The reported timings are averaged over 100 evaluations, following an initial warm-up period of 50 evaluations, and were performed on an NVIDIA A100 GPU.