Beyond Adam: SOAP and Muon for Faster, Label-Efficient Training of Machine Learning Interatomic Potentials

Gil Harari [*1] **Yoel Zimmermann** [*1] **Ola Tangen Kulseng** [1] **Laura Zichi** [1] **Chuin Wei Tan** [1]
Marc L. Descoteaux [1] **Boris Kozinsky** [1 2]

Abstract

Machine learning interatomic potentials (MLIPs) have become a hallmark of AI for scientific simulation. While efforts on new architectures and datasets have led to increasingly accurate and general models, the choice of optimizer for training has largely remained unexplored, defaulting to Adam and its variants in the community. Here, we implement and systematically compare a class of recently proposed matrix-structured optimizers, including Muon, SOAP, and the hybrid SOAP-Muon, for training NequIP and Allegro MLIP models. We find that these optimizers can substantially outperform Adam in both convergence speed and final accuracy. SOAP and SOAP-Muon emerge as robust and consistently strong methods, while Muon only provides partial gains relative to Adam. The improvements are particularly pronounced under partial force supervision. Our results indicate that optimizer choice is an overlooked yet impactful design axis for MLIPs.

1. Introduction

Machine learning interatomic potentials (MLIPs) (Blank et al., 1995; Behler & Parrinello, 2007; Bartók et al., 2010) have become an indispensable tool for atomistic simulation in chemistry and materials science (Husistein & Reiher, 2026). They have enabled studies ranging from the thermodynamics of water (Cheng et al., 2019) to the properties of amorphous silicon (Deringer et al., 2021), large-scale materials screening (Merchant et al., 2023), and even all-atom simulations of complex biomolecular assemblies such as the HIV capsid (Kozinsky et al., 2023).

Motivated by these successes, the MLIP community has invested heavily in two primary directions of improvement. The first is *new architectures*, progressing from descriptor-based (Bartók et al., 2010; Bartók et al., 2013; Shapeev, 2016; Han et al., 2018; Drautz, 2019) to deep learning approaches such as invariant (Schütt et al., 2017) and equivariant graph neural networks (Batzner et al., 2022; Batatia et al., 2022; Musaelian et al., 2023; Bochkarev et al., 2024) and attention-based architectures (Liao & Smidt, 2023; Pozdnyakov & Ceriotti, 2023; Qu & Krishnapriyan, 2024; Qu et al., 2026). The second direction is *new datasets*, with large-scale quantum chemical databases for organic molecules (Smith et al., 2017; Eastman et al., 2023; Levine et al., 2025), catalytic systems (Chanussot et al., 2021), and inorganic materials (Deng et al., 2023; Barroso-Luque et al., 2024) providing the corpora on which these architectures are trained and evaluated.

A third axis of improvement, the *training techniques* that dictate the dynamics of learning, has received comparatively little attention. Crucially, the choice and configuration of the optimizer used is largely static. The vast majority of MLIP training pipelines default to Adam (Kingma & Ba, 2017) or one of its closely related variants such as AdamW (Loshchilov & Hutter, 2019). Even at the scale of training universal potentials, such as the Universal Models for Atoms (UMA) (Wood et al., 2025) and SevenNet-Omni (Kim et al., 2026), Adam-family optimizers remain the default choice in reported training protocols.

Meanwhile, the scaling of large language models (LLMs) has renewed interest in optimizers that go beyond the diagonal gradient scaling of Adam by exploiting matrix structure in neural-network weight tensors. This broad direction includes explicit structured preconditioners such as Shampoo (Gupta et al., 2018), SOAP (Vyas et al., 2025a), Kron (Castanyer et al., 2026), SPlus (Frans et al., 2026), and DyKAF (Yudin et al., 2025); orthogonalized matrix-update methods such as Muon (Jordan et al., 2024), Scion (Pethick et al., 2025), Gluon (Riabinin et al., 2026), and PolarGrad (Lau et al., 2025); and hybrid adaptive schemes such as SOAP-Muon (Vyas et al., 2025b), COSMOS (Liu et al., 2026a), Mousse (Zhang et al., 2026), and Newton-Muon (Du & Su, 2026).

[*]Equal contribution [1]John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA [2]Robert Bosch LLC Research and Technology Center, Watertown, MA, USA. Correspondence to: Boris Kozinsky <bkoz@seas.harvard.edu>.

Accepted at ICML 2026 AI for Science Workshop.

In this work, we focus on **Muon** and **SOAP** as prominent optimizers, together with **SOAP-Muon** as a hybrid variant combining elements of both. For LLMs, these methods have demonstrated faster convergence to competitive loss values (Jordan et al., 2024; Liu et al., 2025; Vyas et al., 2025a). However, their potential for MLIP training remains underexplored. Liu et al. (2026b) recently argue that optimizer choice is an overlooked factor in MLIP fine-tuning, but restrict their benchmark to optimizers based on diagonal preconditioning methods. Koker et al. (2025) provided encouraging evidence for this line of inquiry, showing that Muon outperforms Adam as an ingredient for budget-conscious foundation potential training, though without extensive ablations.

We work towards filling this gap by benchmarking Muon, SOAP, and SOAP-Muon against AdamW on two systems of physical significance: liquid water (Cheng et al., 2019), modeled with NequIP (Batzner et al., 2022) and the solid acid electrolyte CsH_2PO_4 (cesium dihydrogen phosphate, CDP) (Wang et al., 2025a), important for electrochemical energy technologies, modeled with Allegro (Musaelian et al., 2023). We additionally examine optimizer behavior under varying levels of force supervision, including energy-only training. Computing force labels in density functional theory (DFT) incurs little additional cost beyond the energy evaluation, thanks to the Hellmann–Feynman theorem (Helgaker et al., 2013). Higher levels of theory, however, do not share this benefit. For coupled cluster methods, force computation is costly (Smith et al., 2019). This limitation is shared by diffusion Monte Carlo, where energy-only training is typically used due to the prohibitive cost of force evaluations (Huang & Rubenstein, 2022). Methods that are better suited for sparse-force regimes are thus desirable for training more accurate MLIPs on reference data beyond DFT. Furthermore, even when forces are available, some works have trained on only a subset to reduce the training time and memory requirements (López-Zorrilla et al., 2023), highlighting sparse force supervision as a practically relevant regime.

Our contributions here are as follows:

- We integrate Muon, SOAP, and SOAP-Muon into the `nequip` MLIP framework (Tan et al., 2026) and benchmark them against AdamW across two systems. We find that SOAP and SOAP-Muon consistently improve energy and force accuracy and accelerate convergence, with SOAP showing the most robust behavior across systems and SOAP-Muon achieving the strongest results in selected settings. Muon provides gains over AdamW only in one system.
- We examine how these optimizers behave under reduced force supervision: SOAP and SOAP-Muon retain strong accuracy as force supervision is reduced; in particular, SOAP-Muon trained with 50% of force labels matches AdamW trained with 100%, suggesting a path to reducing force-label requirements in regimes where force labels are expensive.
- We demonstrate that the resulting MLIPs are physically faithful, reproducing corresponding *ab initio* calculations and experimental observables. Notably, in one of the systems studied, SOAP-Muon preserves this fidelity even when trained with only 5% of the force labels, while the corresponding AdamW model becomes unstable at the same level of force supervision.

2. Background

MLIP Training MLIPs approximate the potential energy surface (PES) of an atomistic system by fitting a neural network with weights θ to quantum mechanical reference data, most commonly computed using DFT. The network maps a set of atomic positions and chemical species $\{\mathbf{r}_j, Z_j\}$ to a total energy E_θ, which decomposes into a sum of local atomic contributions $\varepsilon_{i,\theta}$ to ensure size extensivity,

$$E_\theta(\{\mathbf{r}_j, Z_j\}) = \sum_i \varepsilon_{i,\theta}(\{\mathbf{r}_j, Z_j\}_{j\in\mathcal{N}_i}), \tag{1}$$

where $\mathcal{N}_i$ denotes the local neighborhood of atom i within a cutoff radius. Forces are obtained as the negative gradient of the predicted energy with respect to atomic positions via automatic differentiation,

$$\mathbf{F}_{i,\theta} = -\frac{\partial E_\theta}{\partial \mathbf{r}_i}, \tag{2}$$

enforcing energy conservation by construction. Training is performed by minimizing a weighted combination of energy and force errors over a dataset of N reference configurations $\{(\{\mathbf{r}_j, Z_j\}_n, E_n, \mathbf{F}_n)\}_{n=1}^N$, where $\mathbf{F}_n \in \mathbb{R}^{3N_{\text{atoms}}}$ collects the reference forces on all atoms in configuration n,

$$\mathcal{L}(\theta) = \frac{\lambda_E}{N}\sum_{n=1}^{N}\left(E_\theta(\{\mathbf{r}_j, Z_j\}_n) - E_n\right)^2 + \frac{\lambda_F}{3\sum_{n=1}^{N} N_{\text{atoms}}^{(n)}}\sum_{n=1}^{N}\left\|\mathbf{F}_\theta(\{\mathbf{r}_j, Z_j\}_n) - \mathbf{F}_n\right\|^2, \tag{3}$$

where λ_E and λ_F control the relative contribution of energy and force supervision. Including forces in the training loss substantially improves accuracy and stability (Batzner et al., 2022). In the force-sparse setting we study here, the force loss is computed over only a subset of configurations, with the limiting case $\lambda_F = 0$ corresponding to energy-only training (see Appendix B for details).

NequIP and Allegro NequIP (Batzner et al., 2022) and Allegro (Musaelian et al., 2023) are $E(3)$-equivariant graph neural networks, in which internal features are geometric tensors that transform under rotations, reflections, and translations, yielding energy and force predictions that respect the fundamental 3D Euclidean symmetries of atomistic systems. Local information is mixed across the network through equivariant tensor products. NequIP propagates features via message-passing between neighboring atoms, while Allegro builds many-body representations through iterated tensor products of atom-pairwise features within a strictly local cutoff. Together, these architectures laid the foundations for the most widely used class of equivariant MLIPs, known for high accuracy and strong data efficiency attributed to the inductive bias of equivariance (Batzner et al., 2022; Musaelian et al., 2023). Both are implemented within the `nequip` software framework (Tan et al., 2026).

Optimizers Vanilla gradient descent updates parameters in the direction of steepest descent. However, the loss surface $\mathcal{L}(\theta)$ is generally anisotropic, i.e., some directions change the loss more than others per unit step. Preconditioning addresses this inconsistency by rescaling the gradient to account for the local geometry of the loss, allowing larger steps in flat directions and smaller steps in steep ones. For a vector-valued parameter w with gradient g, a preconditioned update takes the form

$$w \leftarrow w - \eta P^{-1} g, \tag{4}$$

where P is a positive definite matrix describing the local curvature and η is the learning rate. A natural choice is the Hessian, but forming, storing, and inverting the Hessian is prohibitively expensive at scale. Practical optimizers therefore choose tractable approximations for P. Additionally, most optimizers incorporate momentum, an exponential moving average of past gradients, to smooth noisy updates and accelerate convergence.

Adam (Kingma & Ba, 2017) approximates P with a diagonal matrix $P = \mathrm{diag}(\hat{v}_t)^{1/2}$, where $\hat{v}_t$ is a running estimate of the element-wise squared gradient. **AdamW** (Loshchilov & Hutter, 2019) decouples the weight decay from the Adam gradient update.

For a matrix-valued parameter W with gradient $G \in \mathbb{R}^{m\times n}$, the full preconditioner on $\mathrm{vec}(W)$ would be an $mn \times mn$ matrix, which is intractable in practice. An approximation is to assume Kronecker structure $P \approx L^{\mathsf{T}} \otimes R$, which leads to the update

$$W \leftarrow W - \eta L^{-p} G R^{-p}, \tag{5}$$

where $L \in \mathbb{R}^{m\times m}$ and $R \in \mathbb{R}^{n\times n}$ are left and right preconditioners, and p is the power of the preconditioner.

Shampoo (Gupta et al., 2018) defines the left and right preconditioners using the accumulated uncentered row and column covariances of the gradient matrix, updating them as $L \leftarrow L + GG^{\mathsf{T}}$ and $R \leftarrow R + G^{\mathsf{T}}G$. The preconditioning power is set to $p = 1/4$. Later work by Bernstein & Newhouse (2024) showed that, when the accumulation is removed, Shampoo simplifies to a semi-orthogonal weight update. In this regime, the method can be interpreted as projecting the gradient matrix onto the nearest semi-orthogonal matrix under the Frobenius norm.

Muon (Jordan et al., 2024) builds on this perspective, by first applying a Nesterov-style momentum update and then performing the orthogonalization step using efficient Newton–Schulz iterations.

SOAP (Vyas et al., 2025a) is motivated by the observation that Shampoo is equivalent to running Adafactor (Shazeer & Stern, 2018) in the eigenspace of the preconditioner. That is, the gradient is transformed as $G'_t \leftarrow Q_L^{\mathsf{T}} G Q_R$ where Q_L and Q_R are the eigenvector matrices of the L and R, respectively. SOAP replaces the Adafactor update in this eigenspace with an AdamW step, and then projects the result back to the original parameter space.

SOAP-Muon (Vyas et al., 2025b) extends SOAP by adding a Muon-inspired orthogonalization step after the standard SOAP step.

3. Experimental Setup

To assess the versatility of the studied optimizers, we evaluate them across different equivariant architectures and chemical environments. We consider two representative settings: homogeneous liquid water (Cheng et al., 2019) modeled with NequIP, and solid-state CDP (Wang et al., 2025a) modeled with Allegro.

Our chosen model configurations are based on previously reported parameters (Batzner et al., 2022; Wang et al., 2025a). To ensure a fair comparison across each model, dataset, and optimizer, we use a systematic hyperparameter tuning protocol (see Appendix B for details).

To probe the limits of each optimizer, we additionally consider reduced force supervision, including the energy-only regime. This setting evaluates how these optimizers perform with reduced data. Accordingly, we repeat the tuning protocol across training sets with varying fractions of force-labeled frames—75%, 50%, 10%, 5%, and 0% (energy-only training)—while retaining all energy labels.

We evaluate the resulting models using per-atom energy and force mean absolute error (MAE), as well as training wall-clock time. However, energy and force error metrics alone do not fully capture the quality of an MLIP (Fu et al., 2022). We therefore further assess the physical fidelity of the optimal PES identified by each optimizer by performing molecular dynamics (MD) simulations using the trained

Table 1. Test MAE, reported as mean ± standard deviation across 5 seeds. Results are shown for the joint energy and force prediction task (E+F) and the energy-only task (E). Gray force entries denote quantities not included in the corresponding training objective. Within each metric and dataset, the best and second-best mean values are indicated by **boldface** and underlining, respectively.

Task	Optimizer	CDP (Allegro)		Water (NequIP)	
		E [meV/atom] (↓)	F [meV/Å] (↓)	E [meV/atom] (↓)	F [meV/Å] (↓)
E+F	AdamW	0.628±0.0434	32.2±0.615	0.773±0.0713	25.7±1.44
	Muon	0.581±0.0328	29.6±0.617	1.53±0.317	26.6±2.90
	SOAP	**0.569±0.0694**	29.6±0.305	0.604±0.0105	**20.9±0.698**
	SOAP-Muon	0.582±0.0469	**27.8±0.634**	**0.590±0.0687**	21.0±1.04
E	AdamW	5.16±0.178	503±31.8	3.21±0.272	306±46.1
	Muon	3.30±0.135	281±9.73	5.37±0.813	591±59.1
	SOAP	3.03±0.389	214±25.7	**2.38±0.573**	**236±54.8**
	SOAP-Muon	**2.75±0.163**	**201±18.8**	2.77±0.346	269±24.5

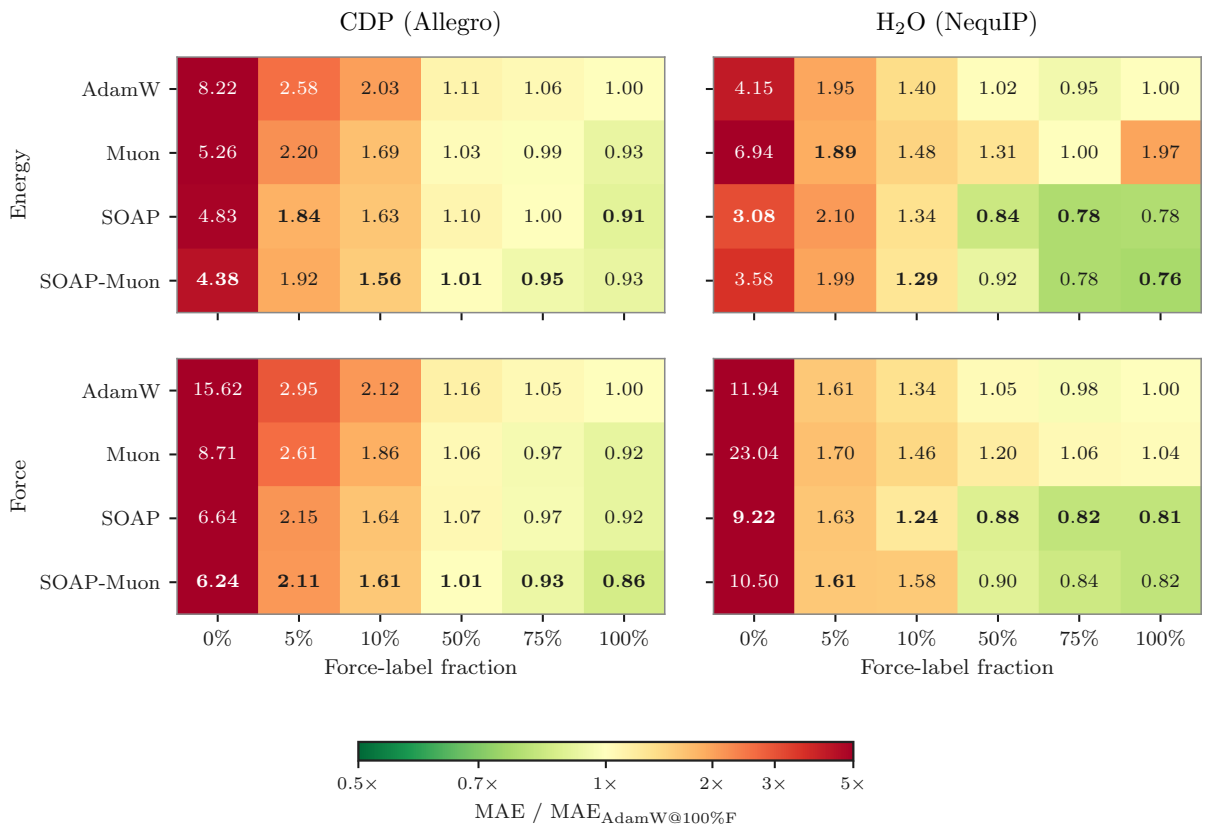

Figure 1. Relative test MAE vs AdamW at full (100%) force supervision. Each cell corresponds to $\text{MAE}_{\text{optimizer}}/\text{MAE}_{\text{AdamW@100\%F}}$. Green cells are below AdamW@100%F (better), red cells above. **Boldface** values indicate the best (lowest) MAE within each force-% column.

MLIPs and comparing resulting observables against experimental measurements and *ab initio* reference data.

4. Results

4.1. Full Supervision

Accuracy Gains Under full energy and force supervision, the matrix-structured optimizers studied substantially outperform AdamW, with the magnitude and consistency of the gains depending on the system (Table 1, Figure 1). On CDP with Allegro, all three optimizers improve over AdamW. SOAP achieves the best energy accuracy, reducing the energy MAE by 9% whereas SOAP-Muon achieves the best force accuracy, reducing the MAE by 14%. On water with NequIP, SOAP and SOAP-Muon again yield lower MAEs than AdamW. In this case, however, SOAP-Muon gives the best energy accuracy, reducing the MAE by 24% while

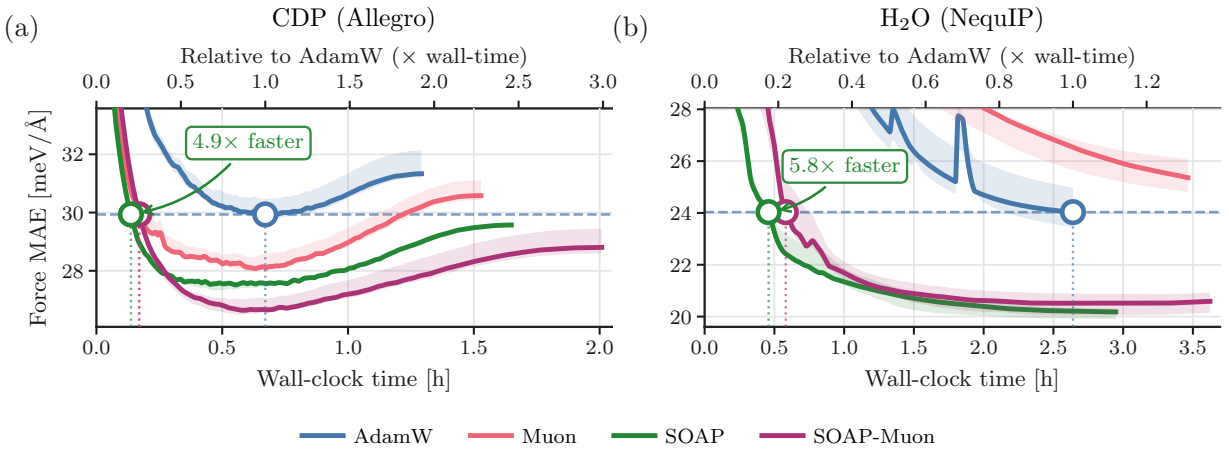

Figure 2. Wall-clock convergence under full force supervision. Validation force MAE versus wall-clock time on CDP (a) and water (b). Lines are per-epoch medians and shaded bands are interquartile ranges across three seeds, all executed on NVIDIA A100 GPUs. The dashed line marks AdamW's minimum median force MAE and circles indicate the first wall-clock time at which each optimizer's median curve crosses that level.

SOAP provides the best force accuracy, reducing the MAE by 19%. Muon alone underperforms the AdamW baseline, particularly for energy prediction.

Speedup The accuracy improvements are accompanied by substantial reductions in time-to-accuracy. We compare the wall-clock time required for each optimizer to reach AdamW's minimum median validation force MAE under full force supervision (Figure 2). On CDP, SOAP reaches that target 4.9× faster than AdamW, while on water it does so 5.8× faster. These gains are notable because matrix-structured methods incur additional computation per optimization step relative to AdamW. The wall-clock advantage therefore implies that they reduce the number of epochs needed by an even larger margin. In practice, improved conditioning more than compensates for the higher per-step cost, making SOAP attractive not only for final accuracy but also for training efficiency.

4.2. Reduced Supervision

Energy-Only Supervision In the energy-only training regime, where forces are excluded from the training objective and therefore provide a stringent test of whether the learned PES captures physically meaningful gradients, the advantage of matrix-structured optimizers becomes more pronounced. On CDP, all three alternatives substantially outperform AdamW. SOAP-Muon achieves the strongest overall results, reducing the energy and force MAE by roughly 47% and 60%. On water, SOAP performs best, reducing both energy and force errors relative to AdamW by 26% and 23%, while Muon performs noticeably worse.

Sparse Force Supervision The same ranking largely persists as the number of force-labeled configurations decreases, with the gains over AdamW becoming especially valuable in the low-label regime (Figure 1). On CDP, SOAP-Muon trained with only 50% force supervision achieves accuracy comparable to AdamW trained with the full force-labeled set, and SOAP shows a similar pattern. Both methods also remain clearly stronger than AdamW at 10% and 5% force supervision. This suggests that improved optimization can partially compensate for reduced access to force labels, lowering the amount of expensive force data needed to reach a given level of accuracy. On water, the trend is less regular, with some models trained at 75% and 50% force supervision often outperforming their respective models trained on 100%. Nevertheless, SOAP is the most consistently competitive method across supervision levels and is the strongest single default in our experiments.

4.3. Physical Fidelity

To assess whether the optimizer-dependent differences in MAE translate into meaningful differences in the learned PESs, we evaluated the trained MLIPs in MD simulations.

CDP We followed the MD protocol of Wang et al. (2025a). All fully force-supervised models faithfully reproduce the *ab initio* MD (AIMD) results, matching the radial distribution functions (RDFs) and showing consistent mean squared displacement (MSD) behaviors across optimizers (Appendix Figure 5). The differences become much more pronounced in the sparse-label regime. Notably, at 5% force supervision, SOAP-Muon remains stable and accurately reproduces the AIMD RDFs and MSD. This is in sharp contrast to AdamW that exhibits catastrophic instability, with trajectories diverg-

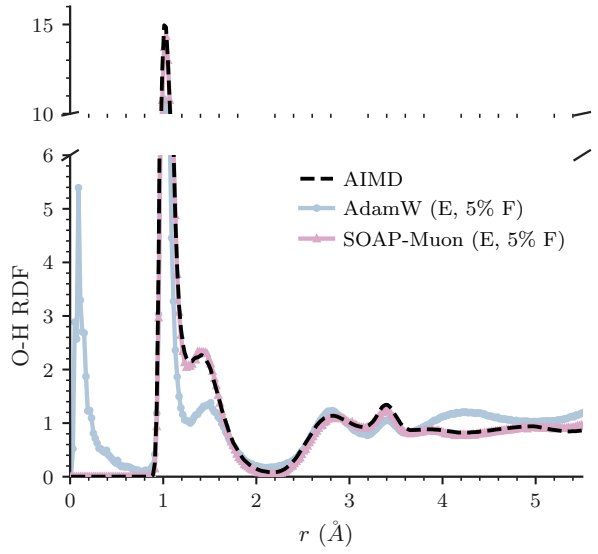

Figure 3. Radial distribution functions (RDFs) of O–H pairs in CDP obtained from MD simulations using AdamW (blue) and SOAP-Muon (pink) models trained on energies and 5% of forces, compared against AIMD ground truth (black).

ing almost immediately and producing nonphysical results under the same conditions, as shown in Figure 3 (the full set of RDFs and MSD is reported in Appendix Figure 7). Moreover, the 5% SOAP-Muon model reproduces experimental results, yielding a proton diffusion activation energy of $E_a = 0.42$ eV (Appendix Figure 7h), in good agreement with the experimental range of $E_a = 0.39$–0.43 eV (Haile et al., 2007; Ishikawa et al., 2008; Wang et al., 2025b).

Water The fully force-supervised models also generate similar structural and dynamical observables across optimizers, with RDFs and MSDs broadly consistent with one another and with the experimental O-O RDF reference (Appendix Figure 6) (Skinner et al., 2014). For all sparsity percentages tested, both AdamW and SOAP maintained physical fidelity on RDFs and MSD (Appendix Figure 8).

5. Discussion

The two cases we study differ both in model architecture (NequIP vs. Allegro) and in chemical character (liquid water vs. multicomponent solid CDP). On CDP, all three matrix-based optimizers outperform AdamW, with a clear progression from AdamW to Muon, SOAP and SOAP-Muon. On water, SOAP and SOAP-Muon also improve upon AdamW, with SOAP having a slight advantage; by contrast, Muon performs consistently worse than AdamW across all force-label fractions tested. We note, however, that unlike CDP, the water case required additional tuning of SOAP-Muon (momentum coefficients and singular-value power; see Appendix B for details) to achieve these results. Taken together, SOAP and SOAP-Muon outperform AdamW across most metrics, training settings, and systems, with SOAP showing the most robust behavior across systems. For practitioners choosing a single default optimizer to use instead of AdamW without further tuning, our results suggest SOAP.

Muon's underperformance on water is a notable result that warrants further investigation. The behavior across all three matrix-structured variants suggests that the orthogonalization step is the primary source of degradation, with adaptive preconditioning mitigating but not eliminating its effects. Similar concerns about the stability of Muon have been raised in the context of physics-informed neural networks (Lu et al., 2026), where step-size regulation along dominant spectral modes was proposed as a potential remedy.

Limitations Our work is restricted to training on two systems and two equivariant architectures. Foundation potential training and fine-tuning are interesting open directions, and we expect SOAP to remain effective at the present scale of foundation models, though this requires a systematic study.

6. Conclusion

Across the systems and training settings studied, matrix-structured optimizers consistently outperform AdamW for MLIP training, with SOAP emerging as the most robust default with the advantage compounding under sparse force supervision. These accuracy gains translate into substantial wall-clock speedups despite per-step preconditioner overhead, and into faithful molecular-dynamics behavior including observables in good agreement with experiment. We argue that training techniques, exemplified by optimizer choice in this work, should be treated as a first-class design axis in MLIP training, alongside architecture and dataset, particularly as the field moves toward universal foundation potentials where label efficiency and convergence speed become increasingly important.

Code Availability

All optimizer code used in this work will be released in future versions of the open-source `nequip` framework and associated repositories, including `github.com/mir-group/nequip` and `github.com/mir-group/allegro`.

Acknowledgments

The authors gratefully acknowledge Menghang Wang for valuable insights and support regarding the CDP calculations. The computations in this paper were run on the FASRC Cannon cluster supported by the FAS Division of Science Research Computing Group at Harvard University. An award of computer time was provided by the INCITE program. This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725. L.Z. was supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-2140743. This work was supported by the National Science Foundation through the Harvard University Materials Research Science and Engineering Center Grant No. DMR-2011754.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Barroso-Luque, L., Shuaibi, M., Fu, X., Wood, B. M., Dzamba, M., Gao, M., Rizvi, A., Zitnick, C. L., and Ulissi, Z. W. Open Materials 2024 (OMat24) Inorganic Materials Dataset and Models. 2024. doi: 10.48550/ARXIV.2410.12771. URL https://arxiv.org/abs/2410.12771.

Bartók, A. P., Payne, M. C., Kondor, R., and Csányi, G. Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons. *Physical Review Letters*, 104(13), apr 1 2010. ISSN 0031-9007. doi: 10.1103/physrevlett.104.136403. URL http://dx.doi.org/10.1103/PhysRevLett.104.136403.

Bartók, A. P., Kondor, R., and Csányi, G. On representing chemical environments. *Physical Review B—Condensed Matter and Materials Physics*, 87(18):184115, 2013.

Batatia, I., Kovacs, D. P., Simm, G., Ortner, C., and Csanyi, G. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 11423–11436. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/4a36c3c51af11ed9f34615b81edb5bbc-Paper-Conference.pdf.

Batzner, S., Musaelian, A., Sun, L., Geiger, M., Mailoa, J. P., Kornbluth, M., Molinari, N., Smidt, T. E., and Kozinsky, B. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13(1), may 4 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-29939-5. URL http://dx.doi.org/10.1038/s41467-022-29939-5.

Behler, J. and Parrinello, M. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Physical Review Letters*, 98(14), apr 2 2007. ISSN 0031-9007. doi: 10.1103/physrevlett.98.146401. URL http://dx.doi.org/10.1103/PhysRevLett.98.146401.

Bernstein, J. and Newhouse, L. Old optimizer, new norm: An anthology, 2024. URL https://arxiv.org/abs/2409.20325.

Bharadwaj, V., Glover, A., Buluç, A., and Demmel, J. An efficient sparse kernel generator for o(3)-equivariant deep networks. In *2025 Proceedings of the Conference on Applied and Computational Discrete Algorithms (ACDA)*, pp. 32–46. Society for Industrial and Applied Mathematics, Philadelphia, PA, January 2025.

Blank, T. B., Brown, S. D., Calhoun, A. W., and Doren, D. J. Neural network models of potential energy surfaces. *The Journal of Chemical Physics*, 103(10):4129–4137, sep 8 1995. ISSN 0021-9606. doi: 10.1063/1.469597. URL http://dx.doi.org/10.1063/1.469597.

Bochkarev, A., Lysogorskiy, Y., and Drautz, R. Graph atomic cluster expansion for semilocal interactions beyond equivariant message passing. *Physical Review X*, 14(2):021036, 2024.

Castanyer, R. C., Obando-Ceron, J., Li, L., Bacon, P.-L., Berseth, G., Courville, A., and Castro, P. S. Stable gradients for stable learning at scale in deep reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=Vqj65VeDOu.

Chanussot, L., Das, A., Goyal, S., Lavril, T., Shuaibi, M., Riviere, M., Tran, K., Heras-Domingo, J., Ho, C., Hu, W., Palizhati, A., Sriram, A., Wood, B., Yoon, J., Parikh, D., Zitnick, C. L., and Ulissi, Z. Open Catalyst 2020 (OC20) Dataset and Community Challenges. *ACS Catalysis*, 11(10):6059–6072, may 4 2021. ISSN 2155-5435. doi: 10.1021/acscatal.0c04525. URL http://dx.doi.org/10.1021/acscatal.0c04525.

Cheng, B., Engel, E. A., Behler, J., Dellago, C., and Ceriotti, M. Ab initio thermodynamics of liquid and solid water. *Proceedings of the National Academy of Sciences*, 116(4): 1110–1115, jan 4 2019. ISSN 0027-8424. doi: 10.1073/pnas.1815117116. URL http://dx.doi.org/10.1073/pnas.1815117116.

Deng, B., Zhong, P., Jun, K., Riebesell, J., Han, K., Bartel, C. J., and Ceder, G. Chgnet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nature Machine Intelligence*, pp. 1–11, 2023. doi: 10.1038/s42256-023-00716-3.

Deringer, V. L., Bernstein, N., Csányi, G., Ben Mahmoud, C., Ceriotti, M., Wilson, M., Drabold, D. A., and Elliott, S. R. Origins of structural and electronic transitions in disordered silicon. *Nature*, 589(7840): 59–64, jan 6 2021. ISSN 0028-0836. doi: 10.1038/s41586-020-03072-z. URL http://dx.doi.org/10.1038/s41586-020-03072-z.

Drautz, R. Atomic cluster expansion for accurate and transferable interatomic potentials. *Phys. Rev. B*, 99:014104, Jan 2019. doi: 10.1103/PhysRevB.99.014104. URL https://link.aps.org/doi/10.1103/PhysRevB.99.014104.

Du, Z. and Su, W. The newton-muon optimizer, 2026. URL https://arxiv.org/abs/2604.01472.

Eastman, P., Behara, P. K., Dotson, D. L., Galvelis, R., Herr, J. E., Horton, J. T., Mao, Y., Chodera, J. D., Pritchard, B. P., Wang, Y., De Fabritiis, G., and Markland, T. E. Spice, A Dataset of Drug-like Molecules and Peptides for Training Machine Learning Potentials. *Scientific Data*, 10(1), jan 4 2023. ISSN 2052-4463. doi: 10.1038/s41597-022-01882-6. URL http://dx.doi.org/10.1038/s41597-022-01882-6.

Frans, K., Levine, S., and Abbeel, P. A stable whitening optimizer for efficient neural network training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=0T8i3uXq3O.

Fu, X., Wu, Z., Wang, W., Xie, T., Keten, S., Gomez-Bombarelli, R., and Jaakkola, T. Forces are not enough: Benchmark and critical evaluation for machine learning force fields with molecular simulations. *arXiv preprint arXiv:2210.07237*, 2022.

Gupta, V., Koren, T., and Singer, Y. Shampoo: Preconditioned stochastic tensor optimization. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1842–1850. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/gupta18a.html.

Haile, S. M., Chisholm, C. R. I., Sasaki, K., Boysen, D. A., and Uda, T. Solid acid proton conductors: from laboratory curiosities to fuel cell electrolytes. *Faraday Discuss.*, 134:17–39, 2007. doi: 10.1039/B604311A. URL http://dx.doi.org/10.1039/B604311A.

Han, J., Zhang, L., Car, R., and E, W. Deep potential: A general representation of a many-body potential energy surface. *Communications in Computational Physics*, 23 (3):629–639, 2018. ISSN 1991-7120. doi: 10.4208/cicp.oa-2017-0213. URL http://dx.doi.org/10.4208/cicp.OA-2017-0213.

Helgaker, T., Jorgensen, P., and Olsen, J. *Molecular electronic-structure theory*. John Wiley & Sons, 2013.

Huang, C. and Rubenstein, B. M. Machine Learning Diffusion Monte Carlo Forces. *The Journal of Physical Chemistry A*, 127(1):339–355, dec 28 2022. ISSN 1089-5639. doi: 10.1021/acs.jpca.2c05904. URL http://dx.doi.org/10.1021/acs.jpca.2c05904.

Husistein, R. T. and Reiher, M. A new paradigm for computational chemistry, 2026. URL https://arxiv.org/abs/2604.01360.

Ishikawa, A., Maekawa, H., Yamamura, T., Kawakita, Y., Shibata, K., and Kawai, M. Proton dynamics of csh2po4 studied by quasi-elastic neutron scattering and pfg-nmr. *Solid State Ionics*, 179(40):2345–2349, 2008. ISSN 0167-2738. doi: https://doi.org/10.1016/j.ssi.2008.10.002. URL https://www.sciencedirect.com/science/article/pii/S0167273808005997.

Jordan, K., Jin, Y., Boza, V., You, J., Cesista, F., Newhouse, L., and Bernstein, J. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Kim, J., You, J., Park, Y., Lim, Y., Kang, Y., Kim, J., Jeon, H., Ju, S., Hong, D., Lee, S. Y., et al. Optimizing cross-domain transfer for universal machine learning interatomic potentials. *Nature Communications*, 2026.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Koker, T., Kotak, M., and Smidt, T. Training a foundation model for materials on a budget, 2025. URL https://arxiv.org/abs/2508.16067.

Kozinsky, B., Musaelian, A., Johansson, A., and Batzner, S. Scaling the Leading Accuracy of Deep Equivariant Models to Biomolecular Simulations of Realistic

Size. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–12. ACM, nov 11 2023. doi: 10.1145/3581784.3627041. URL http://dx.doi.org/10.1145/3581784.3627041.

Lau, T. T.-K., Long, Q., and Su, W. PolarGrad: A class of matrix-gradient optimizers from a unifying preconditioning perspective. *arXiv preprint arXiv:2505.21799*, 2025.

Levine, D. S., Shuaibi, M., Spotte-Smith, E. W. C., Taylor, M. G., Hasyim, M. R., Michel, K., Batatia, I., Csányi, G., Dzamba, M., Eastman, P., Frey, N. C., Fu, X., Gharakhanyan, V., Krishnapriyan, A. S., Rackers, J. A., Raja, S., Rizvi, A., Rosen, A. S., Ulissi, Z., Vargas, S., Zitnick, C. L., Blau, S. M., and Wood, B. M. The Open Molecules 2025 (OMol25) Dataset, Evaluations, and Models. 2025. doi: 10.48550/ARXIV.2505.08762. URL https://arxiv.org/abs/2505.08762.

Liao, Y.-L. and Smidt, T. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=KwmPfARgOTD.

Liu, J., Su, J., Yao, X., Jiang, Z., Lai, G., Du, Y., Qin, Y., Xu, W., Lu, E., Yan, J., Chen, Y., Zheng, H., Liu, Y., Liu, S., Yin, B., He, W., Zhu, H., Wang, Y., Wang, J., Dong, M., Zhang, Z., Kang, Y., Zhang, H., Xu, X., Zhang, Y., Wu, Y., Zhou, X., and Yang, Z. Muon is Scalable for LLM Training, 2025. URL https://arxiv.org/abs/2502.16982.

Liu, L., Xu, Z., Zhang, Z., Kang, H., Li, Z., Liang, C., Chen, W., and Zhao, T. COSMOS: A hybrid adaptive optimizer for efficient training of large language models. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=j2QTOOtM8R.

Liu, X., Wang, Y., and Zhao, T. Beyond Adam: disentangling optimizer effects in the fine-tuning of atomistic foundation models. *AI for Science*, 2(1):015004, mar 1 2026b. ISSN 3050-287X. doi: 10.1088/3050-287x/ae5078. URL http://dx.doi.org/10.1088/3050-287X/ae5078.

López-Zorrilla, J., Aretxabaleta, X. M., Yeu, I. W., Etxebarria, I., Manzano, H., and Artrith, N. ænet-PyTorch: A GPU-supported implementation for machine learning atomic potentials training. *The Journal of Chemical Physics*, 158(16), apr 25 2023. ISSN 0021-9606. doi: 10.1063/5.0146803. URL http://dx.doi.org/10.1063/5.0146803.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Lu, B., Zhang, J., and Lin, G. Muon with Spectral Guidance: Efficient Optimization for Scientific Machine Learning, 2026. URL https://arxiv.org/abs/2602.16167.

Merchant, A., Batzner, S., Schoenholz, S. S., Aykol, M., Cheon, G., and Cubuk, E. D. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, nov 29 2023. ISSN 0028-0836. doi: 10.1038/s41586-023-06735-9. URL http://dx.doi.org/10.1038/s41586-023-06735-9.

Musaelian, A., Batzner, S., Johansson, A., Sun, L., Owen, C. J., Kornbluth, M., and Kozinsky, B. Learning local equivariant representations for large-scale atomistic dynamics. *Nature Communications*, 14 (1), feb 3 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-36329-y. URL http://dx.doi.org/10.1038/s41467-023-36329-y.

Pethick, T., Xie, W., Antonakopoulos, K., Zhu, Z., Silveti-Falls, A., and Cevher, V. Training deep learning models with norm-constrained LMOs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=2Oqm2IzTy9.

Pozdnyakov, S. and Ceriotti, M. Smooth, exact rotational symmetrization for deep learning on point clouds. *Advances in Neural Information Processing Systems*, 36: 79469–79501, 2023.

Qu, E. and Krishnapriyan, A. S. The importance of being scalable: Improving the speed and accuracy of neural network interatomic potentials across chemical domains. *Advances in Neural Information Processing Systems*, 37: 139030–139053, 2024.

Qu, E., Wood, B. M., Krishnapriyan, A. S., and Ulissi, Z. W. A recipe for scalable attention-based mlips: unlocking long-range accuracy with all-to-all node attention. *arXiv preprint arXiv:2603.06567*, 2026.

Riabinin, A., Shulgin, E., Gruntkowska, K., and Richtárik, P. From muon to gluon: Bridging theory and practice of LMO-based optimizers for LLMs, 2026. URL https://openreview.net/forum?id=7aO0YLtXb6.

Schütt, K. T., Kindermans, P.-J., Sauceda, H. E., Chmiela, S., Tkatchenko, A., and Müller, K.-R. Schnet: a continuous-filter convolutional neural network for modeling quantum interactions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*,

NIPS'17, pp. 992–1002, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Shapeev, A. V. Moment tensor potentials: A class of systematically improvable interatomic potentials. *Multiscale Modeling & Simulation*, 14(3):1153–1173, January 2016. ISSN 1540-3467. doi: 10.1137/15m1054183. URL http://dx.doi.org/10.1137/15M1054183.

Shazeer, N. and Stern, M. Adafactor: Adaptive learning rates with sublinear memory cost. *CoRR*, abs/1804.04235, 2018. URL http://arxiv.org/abs/1804.04235.

Skinner, L. B., Benmore, C. J., Neuefeind, J. C., and Parise, J. B. The structure of water around the compressibility minimum. *The Journal of Chemical Physics*, 141(21):214507, 12 2014. ISSN 0021-9606. doi: 10.1063/1.4902412. URL https://doi.org/10.1063/1.4902412.

Smith, J. S., Isayev, O., and Roitberg, A. E. Ani-1, A data set of 20 million calculated off-equilibrium conformations for organic molecules. *Scientific Data*, 4(1), dec 19 2017. ISSN 2052-4463. doi: 10.1038/sdata.2017.193. URL http://dx.doi.org/10.1038/sdata.2017.193.

Smith, J. S., Nebgen, B. T., Zubatyuk, R., Lubbers, N., Devereux, C., Barros, K., Tretiak, S., Isayev, O., and Roitberg, A. E. Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning. *Nature communications*, 10(1):2903, 2019.

Tan, C. W., Descoteaux, M. L., Kotak, M., de Miranda Nascimento, G., Kavanagh, S. R., Zichi, L., Wang, M., Saluja, A., Hu, Y. R., Smidt, T., Johansson, A., Witt, W. C., Kozinsky, B., and Musaelian, A. High-performance training and inference for deep equivariant interatomic potentials. *Digit. Discov.*, 5(4):1558–1567, 2026.

Vyas, N., Morwani, D., Zhao, R., Kwun, M., Shapira, I., Brandfonbrener, D., Janson, L., and Kakade, S. Soap: Improving and stabilizing shampoo using adam, 2025a. URL https://arxiv.org/abs/2409.11321.

Vyas, N., Zhao, R., Morwani, D., Kwun, M., and Kakade, S. Improving soap using iterative whitening and muon, 2025b. URL https://nikhilvyas.github.io/SOAP_Muon.pdf.

Wang, M., Ding, J., Xiong, G., Zhan, N., Owen, C. J., Musaelian, A., Xie, Y., Molinari, N., Adams, R. P., Haile, S., and Kozinsky, B. Revealing the proton slingshot mechanism in solid acid electrolytes through machine learning molecular dynamics, 2025a. URL https://arxiv.org/abs/2503.15389.

Wang, S., Bhartari, A. K., Li, B., and Perdikaris, P. Gradient alignment in physics-informed neural networks: A second-order optimization perspective, 2025b. URL https://arxiv.org/abs/2502.00604.

Wood, B. M., Dzamba, M., Fu, X., Gao, M., Shuaibi, M., Barroso-Luque, L., Abdelmaqsoud, K., Gharakhanyan, V., Kitchin, J. R., Levine, D. S., et al. Uma: A family of universal models for atoms. *arXiv preprint arXiv:2506.23971*, 2025.

Yudin, N., Grishina, E., Veprikov, A., Beznosikov, A., and Rakhuba, M. Dykaf: Dynamical kronecker approximation of the fisher information matrix for gradient preconditioning, 2025. URL https://arxiv.org/abs/2511.06477.

Zhang, Y., Xing, S., Huang, J., Lv, K., Zhou, Y., Qiu, X., Guo, Q., and Chen, K. Mousse: Rectifying the geometry of muon with curvature-aware preconditioning, 2026. URL https://arxiv.org/abs/2603.09697.

A. Optimizer Details

A.1. AdamW

We use the PyTorch `torch.optim.AdamW` implementation.

A.2. Muon

Algorithm 1 summarizes the Muon update for a matrix-valued parameter. The key operation is the orthogonalization of a momentum-augmented gradient matrix. In practice, this orthogonalization is approximated using the Newton–Schulz algorithm (denoted $\mathrm{NewtonSchulz5}(\cdot)$ in Algorithm 1), an iterative matrix procedure for approximating the orthogonal factor $UV^\top$ in the decomposition $G = U\Sigma V^\mathsf{T}$, i.e., the matrix obtained by replacing the singular values of G with ones. Because this orthogonalization is defined for two-dimensional weight tensors (and 4-dimensional convolutional parameters), non-matrix parameters are assigned to an auxiliary AdamW optimizer (for simplicity will be referred to as Adam). Jordan et al. (2024) further observed empirically that embedding and readout weights are also better optimized with Adam.

Our `nequip` implementation builds on the official Muon reference implementation and retains the original β momentum hyperparameter. To accommodate the split between parameters partitioned to Muon and those handled by the auxiliary Adam optimizer, we define two parameter groups, one for each optimizer. Broadly following the recommendations of Jordan et al. (Jordan et al., 2024), we assign intermediate-layer linear weight matrices to Muon, while optimizing the type embeddings and readout layers with Adam. A complete breakdown of the parameter groups is provided in Table 2.

To implement this split in NequIP, we must also account for the internal parameterization of `e3nn` layers. In particular, `e3nn` stores the weights of its equivariant operators as flattened one-dimensional parameter vectors, even when those weights act as structured matrices or tensors in the forward pass. Similar to the Nequix implementation (Koker et al., 2025), for the subset of `e3nn`'s `Linear` parameters assigned to Muon, we first recover the corresponding structured blocks from the flattened representation, perform the Muon update on those reshaped blocks, and then flatten and reassemble them into the original parameter vector. We formalize the instruction-wise action of the `e3nn` operators below.

`e3nn` and tensor-product weights. We consider two `e3nn.o3` layers: (1) `Linear` and (2) `FullyConnectedTensorProduct` (`FCTP`).

In `Linear`, each instruction connects an input irrep block to an output irrep block of the same irrep type. The learnable weight for instruction j is a multiplicity-mixing matrix

$$W_j \in \mathbb{R}^{m_j^{\text{in}} \times n_j^{\text{out}}}, \tag{6}$$

Table 2. Parameter classes, shapes, and optimizer-group assignments for NequIP and Allegro. `e3nn.o3.Linear` (`Linear`) and `e3nn.o3.FullyConnectedTensorProduct` (`FCTP`) layers store weights as flattened 1D parameters, which are partitioned and reshaped into structured blocks during use.

Model	Parameter class	Shape	Parameter group
NequIP	Type embedding	`[num_species, num_features]`	Adam
	Edge-MLP weight matrices	2D weight matrices	Muon
	`Linear` weights	`[weight_numel]` ((m_j, n_j) blocks)	Muon
	`FCTP` weights	`[weight_numel]` ($(m_j^{(1)}, m_j^{(2)}, n_j)$ blocks)	Adam [1]
	Energy readout MLP	`[num_features, 1]`	Adam
	Per-type energy scale and shift	`[num_species, 1]`	Adam
Allegro	Type embeddings	`[num_species, num_tensor_features]`	Adam
	Scalar / tensor embedding MLP weights	2D weight matrices	Muon
	Latent MLP weights	2D weight matrices	Muon
	First-layer environment MLP weights	2D weight matrices	Muon
	Tensor-product weights	`[num_tensor_features, num_paths]`	Adam
	Readout MLP weights	`[num_scalar_features, 1]`	Adam

[1] In future work, these weights could be represented as 2D matrices by flattening the two input dimensions (i.e., $m_j^{(1)} \times m_j^{(2)} \times n_j^{\text{out}} \mapsto \left(m_j^{(1)} m_j^{(2)}\right) \times n_j^{\text{out}}$), analogous to Muon's treatment of convolutional weights.

Algorithm 1 Muon optimizer update (Jordan et al., 2024)

Require: Parameter matrix $W \in \mathbb{R}^{m\times n}$, step size η, β
$M_0 \leftarrow 0$
for $t = 1, 2, \ldots$ **do**
  $G_t \leftarrow \nabla_W \mathcal{L}_t(W_{t-1}) \in \mathbb{R}^{m\times n}$ ▷ Current gradient
  $M_t \leftarrow \beta M_{t-1} + (1-\beta)G_t$ ▷ Momentum buffer
  $U_t \leftarrow \beta M_t + (1-\beta)G_t$ ▷ Nesterov-style momentum
  $O_t \leftarrow \mathrm{NewtonSchulz5}(U_t)$ ▷ Orthogonalized direction
  $W_t \leftarrow W_{t-1} - \eta O_t$ ▷ Parameter update
end for
return W_t

which mixes copies of the irrep but does not act on the internal $2\ell + 1$-dimensional irrep basis.

In `FCTP`, each instruction combines an irrep block from the first input, an irrep block from the second input, and an admissible output irrep block. The learnable weight for instruction j is a multiplicity-mixing tensor

$$T_j \in \mathbb{R}^{m_j^{(1)} \times m_j^{(2)} \times n_j^{\mathrm{out}}}, \tag{7}$$

which mixes copies of the two input irreps and the output irrep but does not act directly on the internal $2\ell + 1$-dimensional irrep bases; those couplings are fixed by the tensor-product Clebsch–Gordan structure.

Thus, `Linear` uses learned 2D multiplicity-mixing blocks, while `FCTP` uses learned 3D blocks over the two input multiplicities and the output multiplicity.

Flattened storage. Although these operators act through structured blocks W_j or T_j, `e3nn` stores all learnable weights as a single flattened parameter vector

$$w \in \mathbb{R}^P, \qquad P = \sum_{j=1}^{J} |\mathbf{s}_j|, \tag{8}$$

where $|\mathbf{s}_j|$ is the number of entries in instruction j's block. Equivalently, we define a slice specification

$$\mathcal{S} = \{(\mathcal{I}_j, \mathbf{s}_j)\}_{j=1}^{J}, \tag{9}$$

with $\mathcal{I}_j$ selecting the entries of w belonging to instruction j, and

$$B_j = \mathrm{reshape}(w_{\mathcal{I}_j}, \mathbf{s}_j), \tag{10}$$

where $B_j = W_j$ and $\mathbf{s}_j = (m_j^{\mathrm{in}}, n_j^{\mathrm{out}})$ for `Linear`, while $B_j = T_j$ and $\mathbf{s}_j = (m_j^{(1)}, m_j^{(2)}, n_j^{\mathrm{out}})$ for `FCTP`. The forward pass uses the reshaped blocks B_j. Adam-style elementwise optimizers can operate directly on the flattened parameter vector w, because their updates do not depend on the structural interpretation of the parameter shape. By contrast, matrix-structured optimizers such as Muon, SOAP, and SOAP-Muon are shape-sensitive. To use their structured orthogonalization or preconditioning, the relevant slices of w must be reshaped into their instruction-wise matrix or tensor blocks, updated in that form, and then flattened back into the original parameter vector.

A.3. SOAP

Algorithm 2 presents the general SOAP update step, simplified by omitting the initialization steps and AdamW weight decay. For the standard SOAP setup, the orthogonalization flag `ortho` and normalization flag `normalize` are turned off. We use the default hyperparameters, $\beta_1 = 0.95$, $\beta_2 = 0.95$, Shampoo $\beta = \beta_2 = 0.95$ and preconditioning frequency of 10 (Vyas et al., 2025a).

Unlike AdamW, whose elementwise updates are largely insensitive to how a parameter tensor is reshaped, SOAP uses Shampoo-style structured preconditioning along tensor modes. Consequently, the tensor shape assigned to each parameter block affects the preconditioner and the resulting update.

Algorithm 2 SOAP update with optional Muon-style orthogonalization and normalization (Vyas et al., 2025a;b).

Require: Matrix block $W \in \mathbb{R}^{m\times n}$, learning rate η, betas $(\beta, \beta_1, \beta_2)$, ϵ, preconditioning frequency f, orthogonalization flag `ortho`, normalization flag `normalize`, singular-value power ρ

for $t = 1, 2, \ldots$ **do**
  $G_t \leftarrow \nabla_W \mathcal{L}_t(W_{t-1}) \in \mathbb{R}^{m\times n}$ ▷ Current gradient
  $\widetilde{G}_t \leftarrow Q_{L,t-1}^{\mathsf{T}} G_t Q_{R,t-1}$ ▷ Gradient projection to Shampoo eigenbasis
  $M_t \leftarrow \beta_1 M_{t-1} + (1-\beta_1)\widetilde{G}_t$ ▷ (Adam) First moment estimate
  $V_t \leftarrow \beta_2 V_{t-1} + (1-\beta_2)(\widetilde{G}_t \odot \widetilde{G}_t)$ ▷ (Adam) Second moment estimate
  $\alpha_t \leftarrow \eta\sqrt{1-\beta_2^t}/(1-\beta_1^t)$ ▷ (Adam) Bias-corrected step size
  $\widetilde{U}_t \leftarrow M_t \oslash (\sqrt{V_t} + \epsilon)$ ▷ (Adam) Elementwise variance-normalization
  $U_t \leftarrow Q_{L,t-1}\widetilde{U}_t Q_{R,t-1}^{\mathsf{T}}$ ▷ Back projection
  if `ortho` **then**
    if $\rho = 0$ **then**
      $U_t \leftarrow \mathrm{NewtonSchulz5}(U_t)$ ▷ (Muon) orthogonalization via Newton–Schulz iteration
    else if $\rho \neq 1$ **then**
      Compute SVD $U_t = P\Sigma R^{\mathsf{T}}$ ▷ (Muon) spectral decomposition
      $U_t \leftarrow P\Sigma^{\rho} R^{\mathsf{T}}$ ▷ (Muon) singular-value power transform
    end if
  end if
  if `normalize` **then**
    $U_t \leftarrow U_t/\sqrt{\mathrm{mean}(U_t^2)}$ ▷ (Muon) RMS normalization
  end if
  $L_t \leftarrow \beta L_{t-1} + (1-\beta) G_t G_t^{\mathsf{T}}$ ▷ Left preconditioner update
  $R_t \leftarrow \beta R_{t-1} + (1-\beta) G_t^{\mathsf{T}} G_t$ ▷ Right preconditioner update
  if $t \,\%\, f = 0$ **then**
    $S_L \leftarrow L_t Q_{L,t-1}$; $S_R \leftarrow R_t Q_{R,t-1}$ ▷ Power iteration
    $Q_{L,t} \leftarrow \mathtt{QR}(S_L)$; $Q_{R,t} \leftarrow \mathtt{QR}(S_R)$ ▷ Update Shampoo eigenbasis
  else
    $Q_{L,t} \leftarrow Q_{L,t-1}$; $Q_{R,t} \leftarrow Q_{R,t-1}$ ▷ Keep previous Shampoo eigenbasis
  end if
  $W_t \leftarrow W_{t-1} - \alpha_t U_t$ ▷ Parameter update
end for
return W_t

To leverage this preconditioning, the `e3nn` linear weights are unstacked and restored to their original two-dimensional forms, following the methodology described in Section A.2. In the present study, the `e3nn` fully connected tensor product weights were not sliced and reshaped into three-dimensional tensors. We leave the implementation of such structural modifications for future investigations.

A.4. SOAP-Muon

Algorithm 2 presents the general SOAP update step with the orthogonalization flag `ortho` and the normalization flag `normalize` both enabled for SOAP-Muon (initialization and AdamW-style weight decay omitted for simplicity). The resulting update combines Muon-style orthogonalization with RMS normalization of the back-projected AdamW update direction. We use the default preconditioner update frequency of 10 and apply full preconditioning, in which the preconditioner is applied along every mode of the weight tensor. Throughout, the Shampoo preconditioner β was set at $\beta = \beta_2$.

A key hyperparameter in this update is the singular-value power ρ, which is applied to the singular-value matrix during the orthogonalization step. The official implementation uses the default value $\rho = 0.5$, which requires a full SVD and is therefore computationally more expensive. A cheaper Muon-style alternative is to set $\rho = 0$, which permits the use of Newton–Schulz iteration (as the transform reduces to UV^{T}). However, Vyas et al. (2025b) reported that using $\rho = 0$ can lead to dataset-dependent training instability. We therefore treat $\rho = 0.5$ as the safer default, while using $\rho = 0$ when training remains stable, since this enables the more efficient Newton–Schulz-based orthogonalization.

Finally, we use the parameter grouping scheme described in Appendix A.2 to determine which parameters receive Muon-style orthogonalization, while the remaining parameters receive standard SOAP updates without the orthogonalization. This grouping also allows us to assign separate β_1, β_2 hyperparameters to the orthogonalized Muon group and the non-orthogonalized Adam group. This differs from the official SOAP-Muon optimizer, which shares the same momentum settings across all parameters.

B. Training Setup

Sparse Force Training Given a desired force sparsity level $s \in [0, 1]$, we retain $N_{\text{keep}} = \text{round}(sN)$ frames with force labels, where N denotes the total number of frames in the corresponding dataset split. In practice, for each sparsity level, the force-labeled frames are sampled uniformly at random once at training initialization. We then define a binary mask $m_n \in \{0, 1\}$ by setting $m_n = 1$ for the retained frames and $m_n = 0$ for the remaining frames, whose force labels are discarded. The same sampled force-labeled subset is used across optimizers for a given data split and random seed.

Under this setting, the loss in Eq. 3 becomes

$$\mathcal{L}(\theta) = \frac{\lambda_E}{N} \sum_{n=1}^{N} \left(E_\theta(\{\mathbf{r}_j, Z_j\}_n) - E_n\right)^2 + \frac{\lambda_F}{3\sum_{n=1}^{N} m_n N_{\text{atoms}}^{(n)}} \sum_{n=1}^{N} m_n \left\|\mathbf{F}_\theta(\{\mathbf{r}_j, Z_j\}_n) - \mathbf{F}_n\right\|^2 . \tag{11}$$

Thus, the force term is normalized only by the number of force components in the force-labeled configurations, rather than by all configurations in the split. The coefficients λ_E and λ_F are kept fixed across sparsity levels using the ratio tuned under full force supervision. The same masking procedure is applied to the training and validation subsets, whereas the test subset is left unmasked.

Model Architectures. The model architectures used in this work are based on previously reported configurations for Allegro on CDP and NequIP on water, which were originally optimized using Adam. The architecture hyperparameters are presented in Table 3.

Hyperparameter Search First, a grid search was performed for the learning rate and weight decay. For each optimizer and task, we first identified a candidate learning-rate order of magnitude 10^{-p}, and then swept over $\{3 \times 10^{-(p+1)}, 10^{-p}, 3 \times 10^{-p}\}$ while weight decay was sampled across four values $\{0, 10^{-5}, 10^{-4}, 10^{-3}\}$. For the water dataset with NequIP, which proved more sensitive to learning rate, we included an additional value of 5×10^{-p}. In Muon and SOAP-Muon, the learning rate and weight decay for both parameter groups were fixed together, although they can be further optimized separately. With these parameters fixed, then, for the full force-supervision setting (100%), the energy-force loss coefficient ratio was tuned by sweeping through $\{1:1, 1:10, 1:100\}$. Final configurations were selected using a validation metric given by a weighted sum of the per-atom energy and force errors, with a 10:1 weighting between the two components.. For NequIP on the water dataset, we used a reduce on plateau scheduler following the settings in Batzner et al. (Batzner et al., 2022). For Allegro on CDP, we opted for a different approach and used a cosine annealing scheduler. All hyperparameter sweeps were run with a single random seed and shortened training schedules: 100 epochs for CDP and 500 epochs for water.

Training Runs The full training runs were conducted using five random seeds (7, 42, 123, 2026, and 1618). The best optimizer hyperparameters identified in the sweeps, and subsequently used in the training runs, are reported in Table 4. Training was run for 1000 epochs for CDP and 2000 epochs for water. Early stopping was applied using the weighted validation objective. The minimum improvement thresholds were 10^{-6} for water and 10^{-5} for CDP, with patience values of 600 and 500 epochs, respectively. Training was conducted on NVIDIA A100-SXM4-80GB and H200 GPUs. To ensure consistency, all wall-clock time comparisons were derived exclusively from training runs performed on A100s without early stopping. All models were compiled (Tan et al., 2026) and accelerated using OpenEquivariance and cuEquivariance tensor-product kernels (Bharadwaj et al., 2025).

SOAP-Muon Stability For the two model–system configurations considered in this work, we observed different sensitivities to the SOAP-Muon hyperparameters: the momentum parameters (β_1, β_2) for each parameter group and the singular-value power ρ applied to the singular-value matrix. Allegro-CDP was relatively robust to these choices, whereas NequIP-water exhibited substantially greater sensitivity and benefited more from further tuning (Figure 4). For the momentum parameters, we searched over $(\beta_1, \beta_2) \in \{(0.9, 0.95), (0.95, 0.95), (0.95, 0.98)\}$, following the combination of values

Table 3. Model architecture hyperparameters used for the CDP and water experiments.

(a) CDP experiments with Allegro (Wang et al., 2025a).

Hyperparameter	Value
Cutoff radius	7.0 Å
Number of layers	2
$l_{\max}$	2
Parity	True
Number of scalar features	32
Number of tensor features	32
Radial Bessel basis functions	8
Trainable Bessel basis	False
Polynomial cutoff exponent	6
Radial-chemical embedding dimension	32
Scalar embedding MLP depth	2
Scalar embedding MLP width	128
Scalar embedding nonlinearity	SiLU
Allegro MLP depth	2
Allegro MLP width	128
Allegro MLP nonlinearity	SiLU
Tensor-product path-channel coupling	False
Readout MLP depth	1
Readout MLP width	32
Readout MLP nonlinearity	None
Per-type energy scales trainable	True
Per-type energy shifts trainable	True

(b) Water experiments with NequIP (Batzner et al., 2022).

Hyperparameter	Value
Cutoff radius	4.5 Å
Number of interaction layers	4
$l_{\max}$	2
Parity	True
Number of features	32
Radial Bessel basis functions	8
Trainable Bessel basis	False
Polynomial cutoff exponent	6
Radial MLP depth	3
Radial MLP width	64
Per-type energy scales trainable	True
Per-type energy shifts trainable	True
Pair potential	ZBL

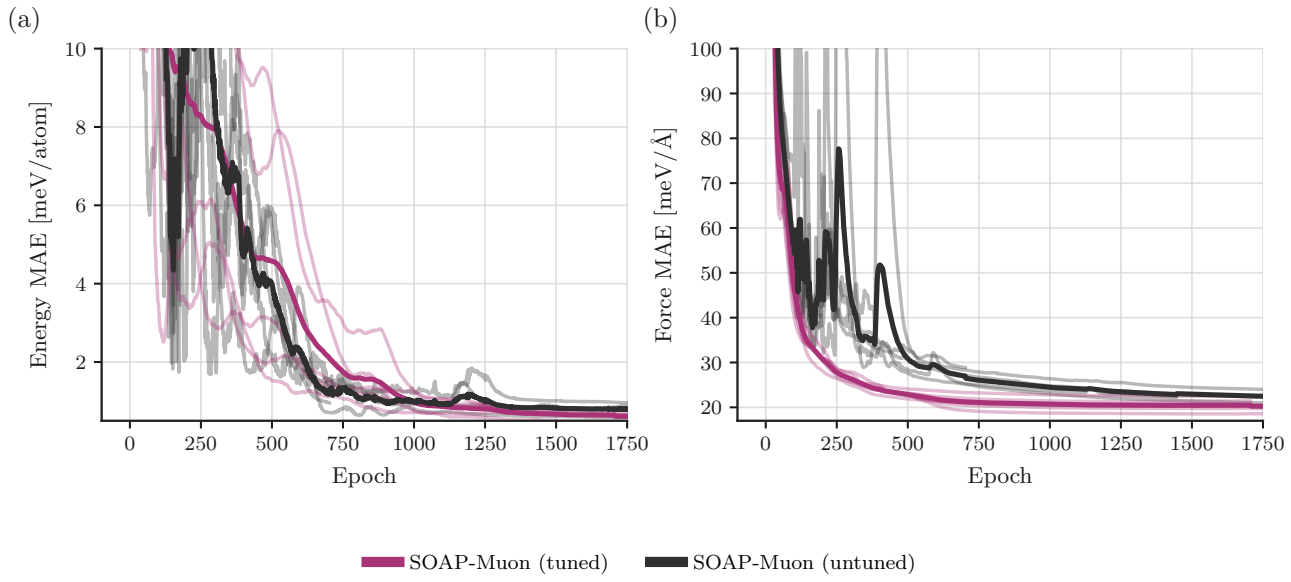

Figure 4. Force and per-atom energy validation training curves for a tuned SOAP-Muon ($\rho = 0.5$, $(\beta_1, \beta_2) = (0.95, 0.95)$ for both parameter groups) and an untuned SOAP-Muon ($\rho = 0.0$, $(\beta_1, \beta_2) = (0.95, 0.98)$ for the Muon parameter group and $(\beta_1, \beta_2) = (0.9, 0.95)$ for the Adam parameter group). Solid lines are per-epoch means, while the faint background lines show each individual seed run. Both tuned and untuned variants use the same learning rate (10^{-3}) and weight decay (10^{-4}).

reported in Jordan et al. (2024); Vyas et al. (2025a;b). For the singular-value power, we considered $\rho \in \{0, 0.5\}$ following Vyas et al. (2025b). We used $\rho = 0$ whenever performance differences were small, in order to avoid the additional cost of the full SVD computation. For Allegro-CDP, we used $(\beta_1, \beta_2) = (0.95, 0.98)$ for the Muon parameter group, $(0.9, 0.95)$ for the Adam parameter group, and $\rho = 0$. For NequIP-water, we used $(\beta_1, \beta_2) = (0.95, 0.95)$ for both the Muon and Adam parameter groups, together with $\rho = 0.5$. Since NequIP-water was more sensitive to these hyperparameters, we treat those settings as a practical default. However, broader evaluations across architectures and datasets are needed to establish more general hyperparameter recommendations. The dataset-specific instability observed for SOAP-Muon is consistent with Vyas et al. (2025b), who found that optimizer stability varied across datasets: some datasets were stable without modification, whereas others required stability-enhancing adjustments. They further showed that datasets stable under the original optimizer were relatively insensitive to these modifications. Nevertheless, our conclusion is based on the limited set of experiments performed here. We note that SOAP-Muon required substantially smaller learning rates than the other optimizers. Learning rates that were optimal for the other optimizers often led to severe training instabilities when used with SOAP-Muon. Throughout our experiments, the best-performing SOAP-Muon learning rate was approximately one order of magnitude lower than those used for the other optimizers (Table 4).

Table 4. Hyperparameter settings for each task, force-% setting, and optimizer.

Task	Force %	Optimizer	CDP (Allegro)		Water (NequIP)	
			Learning rate	Weight decay	Learning rate	Weight decay
E+F	100	AdamW	3×10^{-2}	1×10^{-4}	1×10^{-2}	1×10^{-4}
		Muon	3×10^{-2}	1×10^{-3}	1×10^{-2}	1×10^{-5}
		SOAP	3×10^{-2}	1×10^{-4}	1×10^{-2}	1×10^{-4}
		SOAP-Muon	1×10^{-3}	1×10^{-5}	1×10^{-3}	1×10^{-4}
	75	AdamW	3×10^{-2}	1×10^{-3}	1×10^{-2}	0
		Muon	3×10^{-2}	1×10^{-3}	1×10^{-2}	1×10^{-4}
		SOAP	3×10^{-2}	1×10^{-4}	5×10^{-3}	0
		SOAP-Muon	1×10^{-3}	0	1×10^{-3}	0
	50	AdamW	3×10^{-2}	1×10^{-3}	1×10^{-2}	0
		Muon	3×10^{-2}	1×10^{-3}	5×10^{-3}	0
		SOAP	3×10^{-2}	1×10^{-4}	1×10^{-2}	0
		SOAP-Muon	1×10^{-3}	0	1×10^{-3}	1×10^{-4}
	10	AdamW	1×10^{-2}	1×10^{-3}	1×10^{-2}	1×10^{-5}
		Muon	1×10^{-2}	1×10^{-3}	1×10^{-2}	1×10^{-4}
		SOAP	1×10^{-2}	1×10^{-3}	1×10^{-2}	1×10^{-4}
		SOAP-Muon	1×10^{-3}	1×10^{-4}	1×10^{-3}	0
	5	AdamW	1×10^{-2}	1×10^{-3}	5×10^{-3}	1×10^{-4}
		Muon	1×10^{-2}	1×10^{-3}	1×10^{-2}	1×10^{-4}
		SOAP	1×10^{-2}	1×10^{-5}	5×10^{-3}	1×10^{-3}
		SOAP-Muon	1×10^{-3}	1×10^{-4}	1×10^{-3}	1×10^{-5}
E	0	AdamW	1×10^{-2}	0	5×10^{-3}	1×10^{-3}
		Muon	3×10^{-3}	1×10^{-5}	5×10^{-3}	0
		SOAP	3×10^{-3}	1×10^{-5}	5×10^{-3}	1×10^{-4}
		SOAP-Muon	3×10^{-4}	0	3×10^{-4}	1×10^{-3}

C. Full Results

Table 5. Test MAE aggregated across runs (mean ± std) across 5 seeds. Gray force entries indicate metrics for targets not included in the task focus. **Bold** and underlined entries mark the best and second-best mean values within each force-% setting and metric, respectively.

TASK	FORCE %	OPTIMIZER	CDP (ALLEGRO)		WATER (NEQUIP)	
			E [meV/atom]	F [meV/Å]	E [meV/atom]	F [meV/Å]
E+F	100	ADAMW	0.628±0.0434	32.2±0.615	0.773±0.0713	25.7±1.44
		MUON	0.581±0.0328	29.6±0.617	1.53±0.317	26.6±2.90
		SOAP	**0.569±0.0694**	29.6±0.305	0.604±0.0105	**20.9±0.698**
		SOAP-MUON	0.582±0.0469	**27.8±0.634**	**0.590±0.0687**	21.0±1.04
	75	ADAMW	0.664±0.0383	33.9±0.993	0.738±0.117	25.1±1.34
		MUON	0.622±0.0287	31.1±1.11	0.773±0.146	27.1±1.76
		SOAP	0.630±0.0185	31.3±0.617	**0.601±0.0381**	**20.9±0.872**
		SOAP-MUON	**0.596±0.0280**	**30.1±0.452**	0.602±0.101	21.6±1.27
	50	ADAMW	0.698±0.0175	37.4±0.778	0.789±0.147	26.8±1.30
		MUON	0.645±0.0508	34.1±0.803	1.01±0.153	30.8±2.86
		SOAP	0.690±0.0724	34.5±0.869	**0.650±0.0772**	**22.7±1.01**
		SOAP-MUON	**0.637±0.00589**	**32.5±1.01**	0.713±0.191	23.1±1.07
	10	ADAMW	1.28±0.0809	68.4±2.54	1.09±0.198	34.5±3.69
		MUON	1.06±0.106	60.0±3.10	1.15±0.108	37.4±2.17
		SOAP	1.02±0.0579	52.9±1.65	1.04±0.207	**31.9±2.19**
		SOAP-MUON	**0.977±0.0525**	**51.9±0.842**	**0.999±0.133**	40.6±17.3
	5	ADAMW	1.62±0.0551	94.9±5.09	1.51±0.224	41.3±2.09
		MUON	1.38±0.103	84.0±6.70	**1.46±0.253**	43.5±1.10
		SOAP	**1.16±0.0841**	69.2±1.83	1.62±0.659	41.9±1.59
		SOAP-MUON	1.20±0.0901	**68.1±3.05**	1.54±0.466	**41.2±3.31**
E	0	ADAMW	5.16±0.178	503±31.8	3.21±0.272	306±46.1
		MUON	3.30±0.135	281±9.73	5.37±0.813	591±59.1
		SOAP	3.03±0.389	214±25.7	**2.38±0.573**	**236±54.8**
		SOAP-MUON	**2.75±0.163**	**201±18.8**	2.77±0.346	269±24.5

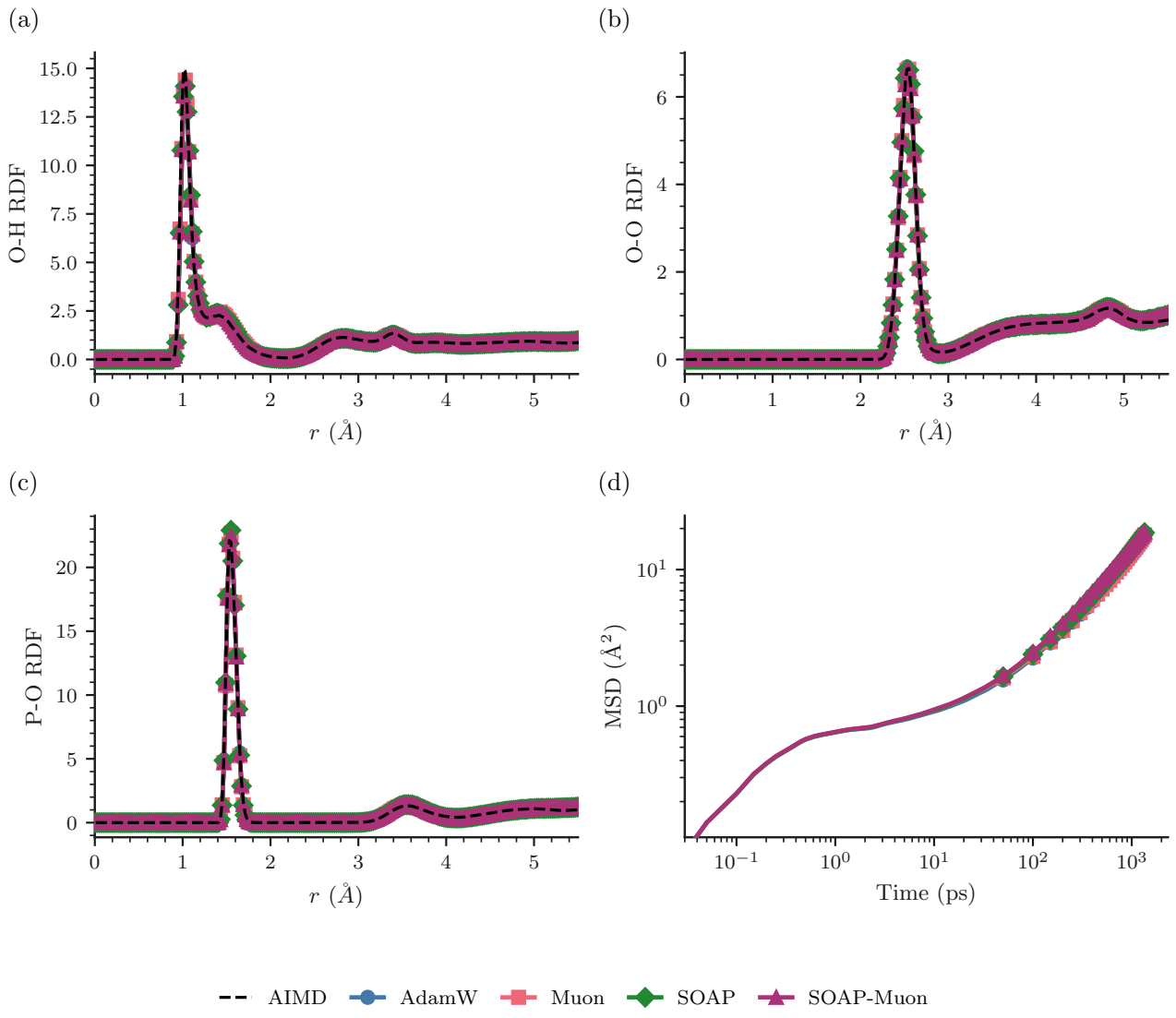

Figure 5. Physical observables obtained from MD simulations of CDP using energy and force trained MLIPs with AdamW (blue), Muon (orange), SOAP (green), and SOAP-Muon (purple), compared against ab initio reference curves. (a–c) Radial distribution functions for the O–H, O–O, and P–O pairs, respectively. (d) Mean squared displacement of H calculated from the corresponding MD trajectories.

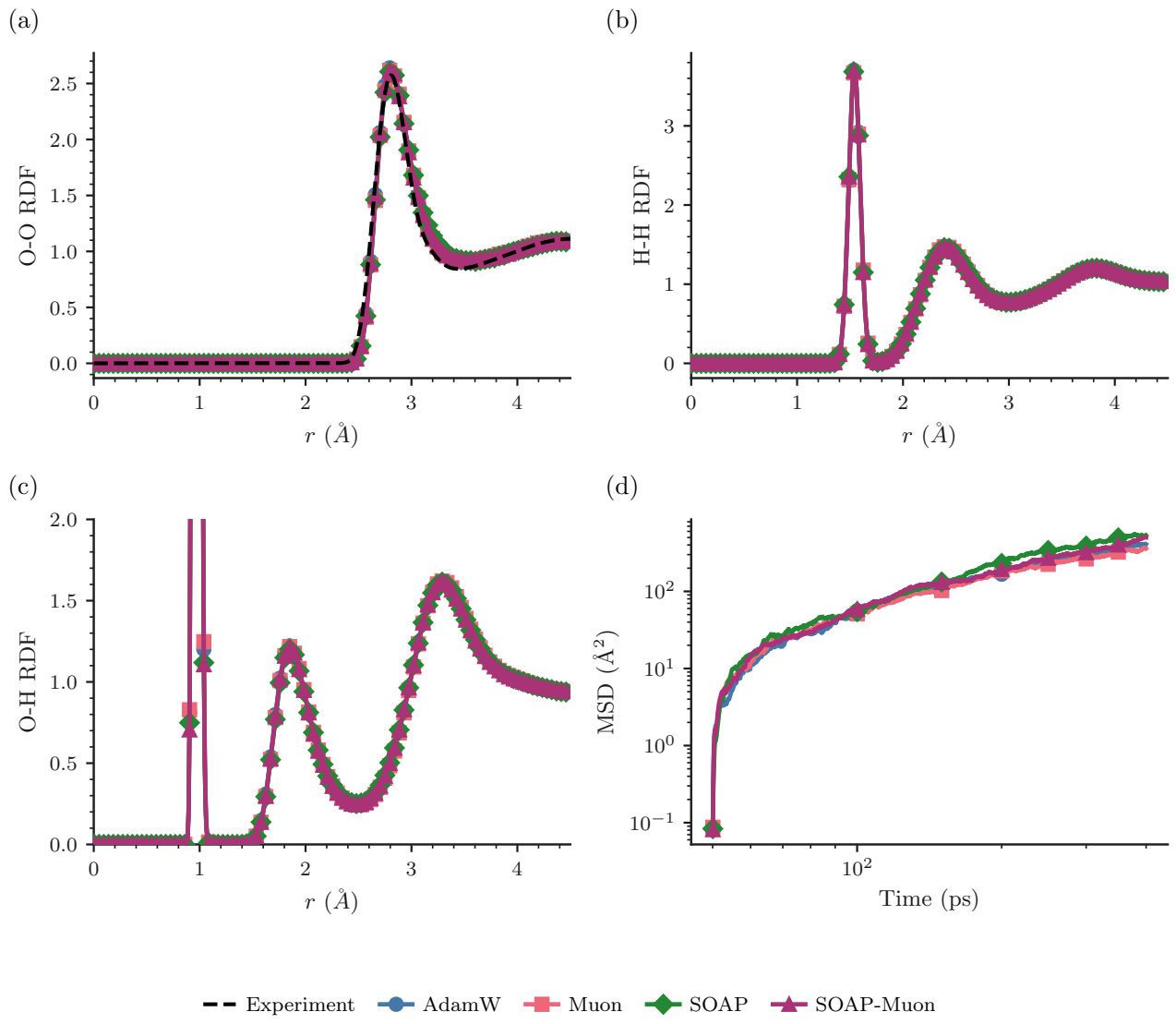

Figure 6. Physical observables obtained from MD simulations of water using energy-and-force-trained MLIPs with AdamW (blue), Muon (orange), SOAP (green), and SOAP-Muon (purple). (a–c) Radial distribution functions (RDFs) for the O–O, H–H, and O–H pairs, respectively. O-O RDF is compared against experimental data from Skinner et al. (2014) (d) Mean squared displacement calculated from the corresponding MD trajectories.

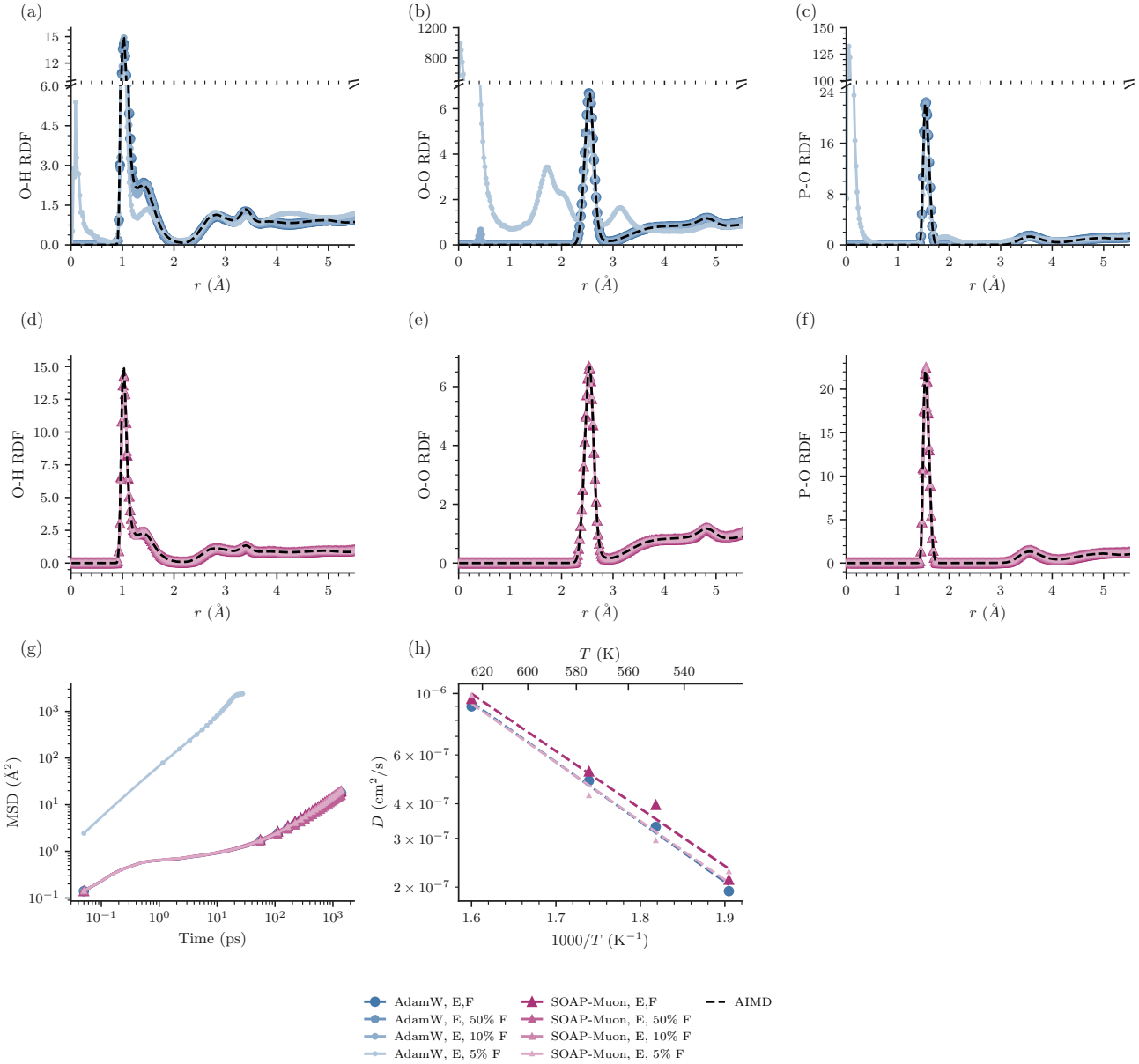

Figure 7. Physical observables obtained from MD simulations of CDP using MLIPs trained with AdamW (blue) and SOAP-Muon (purple), on energy labels and varying fractions of force labels: 5%, 10%, 50%, and 100% force supervision (lighter to darker shades indicate increasing fractions of force labels). (a-c) RDFs for the O–H, O–O, and P–O pairs with AdamW. (d-f) RDFs for the O–H, O–O, and P–O pairs with SOAP-Muon. (g) MSD of H calculated from the corresponding MD trajectories. (h) Arrhenius plot of the diffusion coefficient D, derived from the linear regime of the proton MSD, versus $1/T$. The activation energy is determined from the slope of the fitted linear regression — AdamW (E, F): $E_a = 0.429\,\text{eV}$, SOAP-Muon (E, F): $E_a = 0.411\,\text{eV}$, SOAP-Muon (E, 5% F): $E_a = 0.421\,\text{eV}$. MD simulations with the AdamW (E, 5% F) model did not remain stable long enough to reach the linear proton-diffusion regime. The experimental range is $E_a = 0.39$–0.43 eV (Haile et al., 2007; Ishikawa et al., 2008; Wang et al., 2025a).

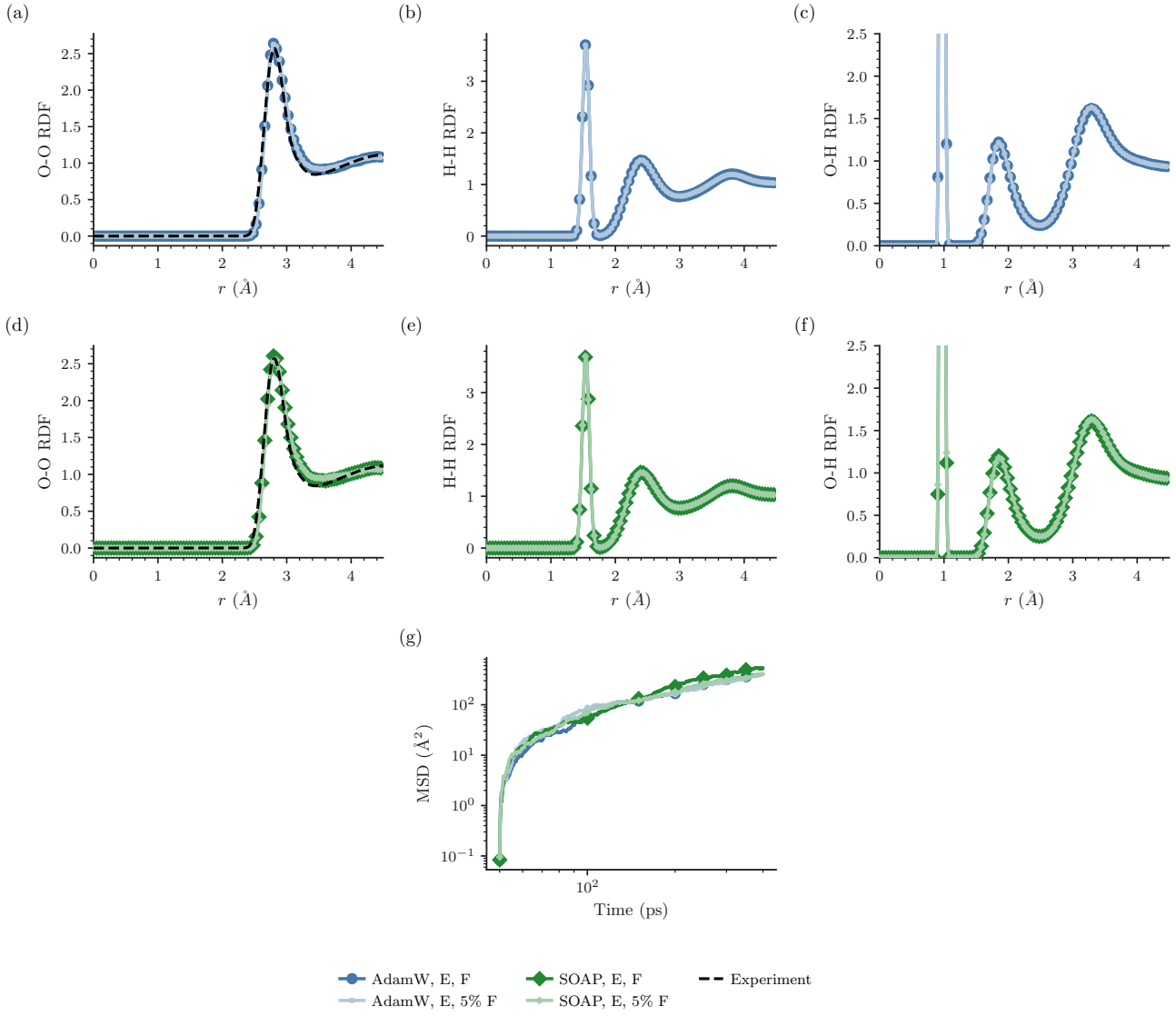

Figure 8. Physical observables obtained from MD simulations of water using MLIPs trained with AdamW (blue) and SOAP (green), on energy labels and varying fractions of force labels: 5% and 100% force supervision (lighter to darker shades indicate increasing fractions of force labels). (a-c) RDFs for the O–O, O–H, and H–H pairs with AdamW. (d-f) RDFs for the O–O, O–H, and H–H pairs with SOAP-Muon. (g) MSD calculated from the corresponding MD trajectories.