Training a Foundation Model for Materials on a Budget

Teddy Koker **Mit Kotak** **Tess Smidt**
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139
{tekoker,mkotak,tsmidt}@mit.edu

Abstract

Foundation models for materials modeling are advancing quickly, but their training remains expensive, often placing state-of-the-art methods out of reach for many research groups. We introduce Nequix, a compact E(3)-equivariant potential that pairs a simplified NequIP design with modern training practices, including equivariant root-mean-square layer normalization and the Muon optimizer, to retain accuracy while substantially reducing compute requirements. Nequix has 700K parameters and was trained in 100 A100 GPU-hours. On the Matbench-Discovery and MDR Phonon benchmarks, Nequix ranks third overall while requiring a 20 times lower training cost than most other methods, and it delivers two orders of magnitude faster inference speed than the current top-ranked model. We release model weights and fully reproducible codebase at `https://github.com/atomicarchitects/nequix`.

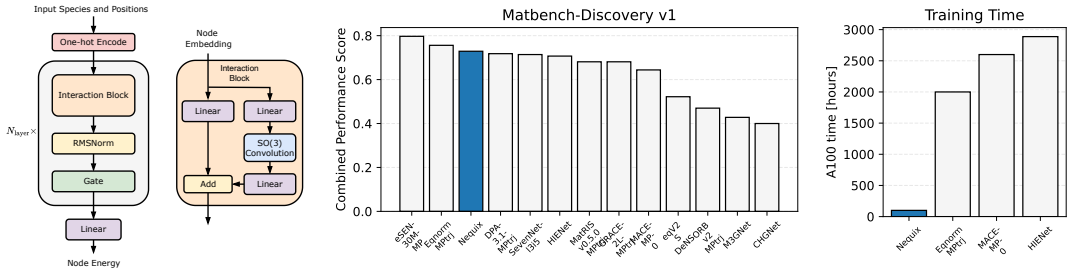
1 Introduction

Machine learned inter-atomic potentials (MLIPs) are rapidly improving in capability and scope, with foundation models trained on broad datasets of atomistic materials offering the promise of augmenting or replacing expensive *ab initio* density functional theory (DFT) calculations [Batatia et al., 2023]. While performance on community benchmarks such as Matbench-Discovery [Riebesell et al., 2025] is rising, the computational costs of both data generation and curation as well as the training of MLIP models on these datasets remain prohibitively expensive for many labs.

We pursue an orthogonal goal to scaling: a lower computational cost recipe that preserves strong downstream accuracy. Concretely, we revisit a simplified E(3)-equivariant architecture based on NequIP [Batzner et al., 2022] with modern training practices: root-mean-square layer normalization for stability, latest custom CUDA kernels [Bharadwaj et al., 2025], and optimizer choices inspired by "speedrunning" deep learning workflows [Jordan et al., 2024a]. The resulting model, Nequix, has 700K parameters and can be trained in 100 GPU hours, while remaining competitive with larger and more costly to train models on Matbench-Discovery and other phonon prediction tasks.

Our contributions are threefold: (1) a simplified NequIP architecture featuring an equivariant layer normalization and efficient JAX and PyTorch implementations; (2) a budget-conscious training pipeline leveraging the Muon optimizer [Jordan et al., 2024b], achieving fast convergence; and (3) evaluations on the Matbench-Discovery and MDR phonon [Loew et al., 2025] benchmarks. Compared to prior MPtrj-trained models [Chen and Ong, 2022, Deng et al., 2023, Batatia et al., 2023, Bochkarev et al., 2024, Neumann et al., 2024, Barroso-Luque et al., 2024, Fu et al., 2025, Zhang et al., 2025, Yan et al., 2025], we rank *third* (as of August 2025) on both benchmarks at 1/20 the training cost of any other published model and with 100$\times$ faster inference than the current top-ranking model.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI4Mat.



(a) (b) (c)

Figure 1: (a) Nequix architecture, a simplified version of NequIP [Batzner et al., 2022], with a species-independent residual connection and layer normalization. (b) Combined performance scores of compliant models on the Matbench-Discovery (unique prototypes subset), collected on 2025-08-17. (c) Available published training times of current compliant models.

2 Methods

Architecture Nequix follows a simplified version of the NequIP [Batzner et al., 2022] architecture, as shown in figure 1a. We adopt two modifications suggested by Park et al. [2024]: the species-specific self-connection layer within the interaction block is replaced with a single linear layer, and unused non-scalar representations are discarded from the final layer. Lastly, we add an equivariant root-mean-square layer normalization (RMSNorm) Liao et al. [2023], which we find improves performance in our optimization setting. We document the full architecture hyper-parameters and the rationale behind each decision in Table A.1.

Implementation Nequix is implemented in both JAX [Bradbury et al., 2018, Kidger and Garcia, 2021] and PyTorch[Paszke et al., 2019, Ansel et al., 2024], taking advantage of just-in-time compilation and efficient automatic differentiation. Following standard energy-conserving MLIP practice [Fu et al., 2025], forces are obtained as the negative energy gradient with respect to atomic positions, $-\nabla_{\mathbf{r}} E$, and stresses as the energy derivative with respect to strain, normalized by volume, $\sigma = V^{-1}\, \partial E / \partial \varepsilon$, where E is the predicted total energy of the system, $\mathbf{r}$ is an atom position, ε is the strain tensor, V is the simulation cell volume, and σ denotes the stress tensor.

Optimization and normalization We compare the widely used Adam optimizer Kingma and Ba [2014] with the recently proposed Muon optimizer Jordan et al. [2024b], which uses the Newton-Schulz algorithm to orthogonalize weight updates. Using a smaller version of Nequix on MPtrj [Jain et al., 2013, Deng et al., 2023] with hidden irreps of `128x0e + 64x1o`, we sweep learning rates of $\{0.03, 0.01, 0.003, 0.001\}$ for Adam and Muon, each with and without RMSNorm. The lowest validation error runs for each optimizer are shown in Fig. 2. We find that the Muon configuration achieves comparable energy/force errors to Adam in 60-70% of the epochs, and results in a 7% reduction in energy MAE. We also find a significant reduction in the variance of stress error, which we notice in runs that use RMSNorm. Notably, the presence of the RMSNorm layer generally resulted in lower validation error for Muon-based training configurations, and higher for those using Adam.

GPU kernels There has been recent work [Bharadwaj et al., 2025, NVIDIA, 2024, Tan et al., 2025, Lee et al., 2025] on writing custom GPU kernels for the expensive equivariant tensor product operation [Xie et al., 2025]. These kernels fuse the tensor product and the outer gather-scatter from the message passing step into a single GPU kernel. This avoids storing costly edge-based intermediates in GPU memory, improving both runtime and memory usage.

Training procedure The final Nequix model is trained for 100 epochs on MPtrj [Jain et al., 2013, Deng et al., 2023], of which we hold out 5% for validation. More details on the training settings are provided in Sec. A.1. The model was trained on 2 NVIDIA A100 80 GB GPUs in 50 hours, for a total cost of 100 GPU hours.

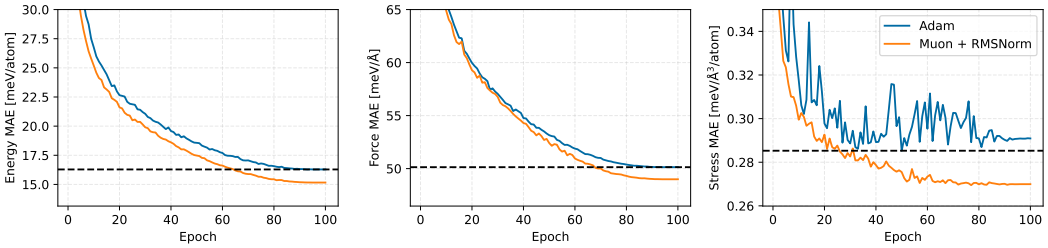


Figure 2: Validation metrics during training of a smaller version of Nequix configuration with Adam and Muon, trying learning rates in $\{0.03, 0.01, 0.003, 0.001\}$ and with/without RMSNorm. This model configuration uses the same hyperparameters as the final model, except with hidden irreps of `128x0e + 64x1o`. The dotted horizontal line shows the best validation performance reached during the Adam training.

3 Experiments

3.1 Matbench-Discovery benchmark

Matbench-Discovery [Riebesell et al., 2025] provides a standard framework for evaluating interatomic potentials in a high-throughput materials screening task consisting of geometry optimization and energy prediction on a set of 257,487 generated structures, and thermal conductivity prediction on a set of 103 structures. Ground truth is calculated with DFT/PBE level of theory, the same as MPtrj. The primary metrics include: 1) the F1 for stable/unstable classification after relaxation; 2) root mean squared displacement (RMSD) between predicted and reference structures after relaxation; and 3) symmetric relative mean error in predicted phonon mode contributions to thermal conductivity κ (κ_{SRME}). A normalized and weighted combination of these metrics are then used to compute a combined performance score (CPS-1), which is used for ranking.

Following Riebesell et al. [2025], we integrate our interatomic potential as Atomic Simulation Environment (ASE) calculator, which is then used to perform structure relaxation and phonon calculations with the default settings of the benchmark. For comparison, we consider only models in the compliant subset of the benchmark. This consists only of models that are trained on MPtrj or subsets, which limits data leakage and offers a more fair comparison among methods. Table 1 contains the performance of Nequix along with all current compliant models at the time of writing. We also include the reported training cost for the models when available, visualized in Fig. 1. We find that Nequix ranks third by CPS-1, outperforming most models at a fraction of the training cost. It is noteworthy that this high ranking is due to high performance in the thermal conductivity task, however, the F1 score is still comparable to many of the other methods.

Table 1: Matbench-Discovery v1 compliant leaderboard, sorted by combined performance score (CPS-1). Metrics are shown for the unique prototypes subset. Train cost is measured in A100 hours. Data as of 2025-08-17.

Model	Params	Train cost	RMSD↓	κ_{SRME} ↓	F1↑	CPS-1↑
eSEN-30M-MP	30.1M	-	0.075	0.340	0.831	0.797
Eqnorm MPtrj	1.31M	2000	0.084	0.408	0.786	0.756
Nequix	708K	100	0.085	0.446	0.750	0.729
DPA-3.1-MPtrj	4.81M	-	0.080	0.650	0.803	0.718
SevenNet-l3i5	1.17M	-	0.085	0.550	0.760	0.714
HIENet	7.51M	2888	0.080	0.642	0.777	0.707
MatRIS v0.5.0 MPtrj	5.83M	-	0.077	0.861	0.809	0.681
GRACE-2L-MPtrj	15.3M	-	0.090	0.525	0.691	0.681
MACE-MP-0	4.69M	2600	0.092	0.647	0.669	0.644
eqV2 S DeNS	31.2M	-	0.076	1.676	0.815	0.522
ORB v2 MPtrj	25.2M	-	0.101	1.725	0.765	0.470
M3GNet	228K	-	0.112	1.412	0.569	0.428
CHGNet	413K	-	0.095	1.717	0.613	0.400

3.2 MDR phonon benchmark

Performance is also evaluated on the MDR phonon benchmark [Loew et al., 2025], a set of 10,000 phonon calculations also done with DFT/PBE level of theory. We follow the identical procedure to Loew et al. [2025], first performing a geometry relaxation, then phonon calculations using displacements of supercells. We report the mean absolute error (MAE) of properties derived from the phonon calculation: maximum phonon frequency $\omega_{\max}$, vibrational entropy S, Helmholtz free energy F, and heat capacity at constant volume C_V. Table 2 demonstrates the performance of Nequix compared to other MPtrj-trained models. Similarly to Matbench-Discovery, we achieve performance within the top three of models, with a fraction of the parameter count of other methods.

Table 2: Model performance of MPtrj-trained models on the MDR phonon benchmark, sourced from Loew et al. [2025] and Fu et al. [2025]. Metrics are MAE of maximum phonon frequency $\omega_{\max}$ (K), vibrational entropy S (J/K/mol), Helmholtz free energy F (kJ/mol) and heat capacity at constant volume C_V (J/K/mol).

Model	MAE($\omega_{\max}$)	MAE(S)	MAE(F)	MAE(C_V)
eSEN-30M	21	13	5	4
SevenNet-l3i5	26	28	10	5
Nequix	26	33	12	6
SevenNet-0	40	48	19	9
GRACE-2L (r6)	40	25	9	5
MACE	61	60	24	13
CHGNet	89	114	45	21
M3GNet	98	150	56	22

3.3 Inference speed

Finally, we compare inference speed with existing interatomic potentials by using ASE Larsen et al. [2017] to run calculations on diamond across varying unit cell sizes, timing the calculations for each model.[1]. We run each model in the default configuration in which it used within its ASE calculator, with compilation and kernels wherever specified in the documentation. See Section A.2 for more information on benchmarking setup.

Figure 3 compares performance of each model in terms of steps per day vs. number of atoms. In this study, Nequix is about 100× faster than eSEN, offering a new option in the accuracy vs. speed Pareto frontier at a fraction of the training cost.

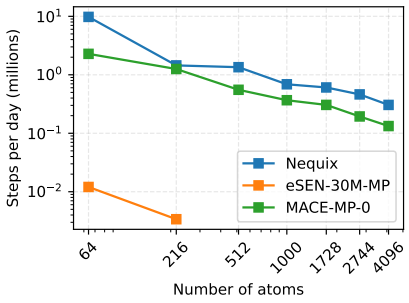


Figure 3: Inference speed of various models in steps per day.

4 Conclusion

We presented Nequix, an E(3)-equivariant interatomic potential that pairs a simplified NequIP architecture with modern training practices. Our results show that Nequix achieves competitive accuracy on Matbench-Discovery and the MDR phonon benchmark at less than one quarter of the reported training cost of many contemporaries. This resource-efficient recipe provides a practical alternative to large-scale foundation models and helps broaden access to high-quality atomistic modeling in settings with more limited compute. We release trained weights and a JAX/PyTorch codebase to streamline reuse and extension.

Looking ahead, we see several promising directions: scaling training duration and data while maintaining budget discipline, exploring pretraining and fine-tuning regimes across broader datasets, and pushing cost even lower through model distillation, pruning, quantization, kernel implementations, or more data-efficient training. We hope Nequix serves as a strong, efficient baseline for future work on accessible materials foundation models.

[1] See `https://github.com/mitkotak/matbench-speed` for inference speed benchmarking code.

Acknowledgments and Disclosure of Funding

This work was supported by the National Science Foundation under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, `http://iaifi.org/`) NSF Graduate Research Fellowship program under Grant No. DGE-1745302, and by DOE ICDI grant DE-SC0022215. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a Department of Energy User Facility using NERSC award ERCAP0033254.

References

Ilyes Batatia, Philipp Benner, Yuan Chiang, Alin M Elena, Dávid P Kovács, Janosh Riebesell, Xavier R Advincula, Mark Asta, Matthew Avaylon, William J Baldwin, et al. A foundation model for atomistic materials chemistry. *arXiv preprint arXiv:2401.00096*, 2023.

Janosh Riebesell, Rhys EA Goodall, Philipp Benner, Yuan Chiang, Bowen Deng, Gerbrand Ceder, Mark Asta, Alpha A Lee, Anubhav Jain, and Kristin A Persson. A framework to evaluate machine learning crystal stability predictions. *Nature Machine Intelligence*, 7(6):836–847, 2025.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):2453, 2022.

Vivek Bharadwaj, Austin Glover, Aydin Buluc, and James Demmel. *An Efficient Sparse Kernel Generator for O(3)-Equivariant Deep Networks*. Society for Industrial and Applied Mathematics, 2025. URL `https://arxiv.org/abs/2501.13986`.

Keller Jordan, Jeremy Bernstein, Brendan Rappazzo, @fernbear.bsky.social, Boza Vlado, You Jiacheng, Franz Cesista, Braden Koszarsky, and @Grad62304977. modded-nanogpt: Speedrunning the nanogpt baseline, 2024a. URL `https://github.com/KellerJordan/modded-nanogpt`.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024b. URL `https://kellerjordan.github.io/posts/muon/`.

Antoine Loew, Dewen Sun, Hai-Chen Wang, Silvana Botti, and Miguel AL Marques. Universal machine learning interatomic potentials are ready for phonons. *npj Computational Materials*, 11(1):178, 2025.

Chi Chen and Shyue Ping Ong. A universal graph deep learning interatomic potential for the periodic table. *Nature Computational Science*, 2(11):718–728, 2022.

Bowen Deng, Peichen Zhong, KyuJung Jun, Janosh Riebesell, Kevin Han, Christopher J Bartel, and Gerbrand Ceder. Chgnet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nature Machine Intelligence*, 5(9):1031–1041, 2023.

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

Mark Neumann, James Gin, Benjamin Rhodes, Steven Bennett, Zhiyi Li, Hitarth Choubisa, Arthur Hussey, and Jonathan Godwin. Orb: A fast, scalable neural network potential. *arXiv preprint arXiv:2410.22570*, 2024.

Luis Barroso-Luque, Muhammed Shuaibi, Xiang Fu, Brandon M Wood, Misko Dzamba, Meng Gao, Ammar Rizvi, C Lawrence Zitnick, and Zachary W Ulissi. Open materials 2024 (omat24) inorganic materials dataset and models. *arXiv preprint arXiv:2410.12771*, 2024.

Xiang Fu, Brandon M Wood, Luis Barroso-Luque, Daniel S Levine, Meng Gao, Misko Dzamba, and C Lawrence Zitnick. Learning smooth and expressive interatomic potentials for physical property prediction. *arXiv preprint arXiv:2502.12147*, 2025.

Duo Zhang, Anyang Peng, Chun Cai, Wentao Li, Yuanchang Zhou, Jinzhe Zeng, Mingyu Guo, Chengqian Zhang, Bowen Li, Hong Jiang, et al. Graph neural network model for the era of large atomistic models. *arXiv preprint arXiv:2506.01686*, 2025.

Keqiang Yan, Montgomery Bohde, Andrii Kryvenko, Ziyu Xiang, Kaiji Zhao, Siya Zhu, Saagar Kolachina, Doğuhan Sarıtürk, Jianwen Xie, Raymundo Arróyave, et al. A materials foundation model via hybrid invariant-equivariant architectures. *arXiv preprint arXiv:2503.05771*, 2025.

Yutack Park, Jaesun Kim, Seungwoo Hwang, and Seungwu Han. Scalable parallel algorithm for graph neural network interatomic potentials in molecular dynamics simulations. *Journal of chemical theory and computation*, 20(11):4857–4868, 2024.

Yi-Lun Liao, Brandon Wood, Abhishek Das, and Tess Smidt. Equiformerv2: Improved equivariant transformer for scaling to higher-degree representations. *arXiv preprint arXiv:2306.12059*, 2023.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/jax-ml/jax`.

Patrick Kidger and Cristian Garcia. Equinox: neural networks in JAX via callable PyTrees and filtered transformations. *Differentiable Programming workshop at Neural Information Processing Systems 2021*, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. URL `https://arxiv.org/abs/1912.01703`.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, C. K. Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Shunting Zhang, Michael Suo, Phil Tillet, Xu Zhao, Eikan Wang, Keren Zhou, Richard Zou, Xiaodong Wang, Ajit Mathews, William Wen, Gregory Chanan, Peng Wu, and Soumith Chintala. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS '24, page 929–947, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703850. doi: 10.1145/3620665.3640366. URL `https://doi.org/10.1145/3620665.3640366`.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, et al. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL materials*, 1(1), 2013.

NVIDIA. cuequivariance: Cuda-accelerated library for equivariant neural networks, 2024. URL `https://github.com/NVIDIA/cuEquivariance`.

Chuin Wei Tan, Marc L. Descoteaux, Mit Kotak, Gabriel de Miranda Nascimento, Seán R. Kavanagh, Laura Zichi, Menghang Wang, Aadit Saluja, Yizhong R. Hu, Tess Smidt, Anders Johansson, William C. Witt, Boris Kozinsky, and Albert Musaelian. High-performance training and inference for deep equivariant interatomic potentials, 2025. URL `https://arxiv.org/abs/2504.16068`.

Seung Yul Lee, Hojoon Kim, Yutack Park, Dawoon Jeong, Seungwu Han, Yeonhong Park, and Jae W. Lee. Flashtp: Fused, sparsity-aware tensor product for machine learning interatomic potentials. In *Proceedings of the 42nd International Conference on Machine Learning*. PMLR, 2025. URL `https://openreview.net/forum?id=wiQe95BPaB`.

YuQing Xie, Ameya Daigavane, Mit Kotak, and Tess Smidt. The price of freedom: Exploring expressivity and runtime tradeoffs in equivariant tensor products, 2025. URL `https://arxiv.org/abs/2506.13523`.

Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N Groves, Bjørk Hammer, Cory Hargus, et al. The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27):273002, 2017.

A Appendix

A.1 Training and model configuration

Table A.1 shows the hyper-parameters used to train Nequix. The model is trained for 100 epochs, using an MAE loss function on energy and stress, and l_2 loss on forces. We use a linear warmup with cosine decay learning rate schedule. Figure A.1 shows the energy, force, and stress MAE on the validation set throughout training. The final MAEs are 10.05 meV/atom, 32.79 meV/Å, and 0.22 meV/Å^3/atom for energy, forces, and stress respectively.

Table A.1: Hyper-parameters used and rationale behind selection

Hyper-parameter	Value	Notes/Rationale
Radial cutoff	6 Å	Most models use 5 or 6 Å; 6 performed slightly better in preliminary validation performance.
Hidden irreps	`128x0e + 64x1o + 32x2e + 32x3o`	From SevenNet-l3i5.
L_{max}	3	Consistent with hidden irreps.
N_{layers}	4	Balance of performance and efficiency.
Radial basis size	8	From NequIP and analysis from Sec. 5.2 of Fu et al. [2025]
Radial MLP size	64	From NequIP.
Radial MLP layers	2	From NequIP.
Polynomial cutoff p	6.0	From NequIP.
Radial basis function	Bessel	From NequIP. Also tried Gaussian, which had minimal difference on validation performance.
Learning rate	0.01	Selected from $\{0.03, 0.01, 0.003, 0.001\}$ based on validation performance early in training.
Warmup epochs	0.1	From eSEN.
Warmup factor	0.2	From eSEN.
Optimizer	Muon	See Sec. 2.
Weight decay	0.001	From eSEN. Also tried 0.0, which led to worse validation performance.
Energy weight	20	From eSEN.
Force weight	20	From eSEN.
Stress weight	5	From eSEN.
Batch size	256 (dynamic)	See Sec. 2
Number of epochs	100	Standard training duration.

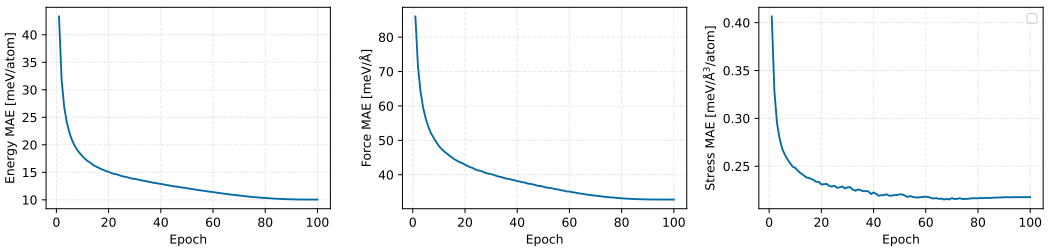


Figure A.1: Validation curves for Nequix training on MPtrj.

A.2 Inference Configuration

We summarize all of the inference configurations in Table A.2. We use MPTrj versions for all of the models except NequIP where the MPtrj model did not have kernel support, so we instead use the OMat24 model after confirming that they have the same hyperparams. For compatibility with eSEN, we use the older `fairchem` OCP calculator, and expect the latest version to have less overhead.

Table A.2: Inference configuration for the ASE benchmarking setup

Model	Compile	Kernels
eSEN-30M	$\times$	$\times$
MACE-MP0	`torch.compile`	cuEquivariance
Nequix	`torch.compile`	openEquivariance